SOFTWARE INTEGRATION FOR AUTOMATED STABILITY ANALYSIS AND DESIGN OPTIMIZATION OF A BEARINGLESS ROTOR BLADE

A Thesis
Presented to
The Academic Faculty

by

Mustafa Emre Gündüz

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Daniel Guggenheim School of Aerospace Engineering

Georgia Institute of Technology
May 2010

SOFTWARE INTEGRATION FOR AUTOMATED STABILITY ANALYSIS AND DESIGN OPTIMIZATION OF A BEARINGLESS ROTOR BLADE

Approved by:

Professor Daniel P. Schrage, Advisor
Daniel Guggenheim School of
Aerospace Engineering
Georgia Institute of Technology

Professor Olivier A. Bauchau
Daniel Guggenheim School of
Aerospace Engineering
Georgia Institute of Technology

Professor Lakshmi N. Sankar
Daniel Guggenheim School of
Aerospace Engineering
Georgia Institute of Technology

Associate Professor Mark Costello
Daniel Guggenheim School of
Aerospace Engineering
Georgia Institute of Technology

Assistant Professor Adeel Khalid
Division of Systems Engineering
Southern Polytechnic State University

Date Approved: 02 April 2010

To my parents

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Daniel Schrage for his generosity and willingness to guide me despite his busy schedule. He provided me a great degree of freedom and flexibility in my research. He was always positive and encouraging. I also thank my thesis committee members Dr. Mark Costello, Dr. Olivier Bauchau, Dr. Lakshmi Sankar and Dr. Adeel Khalid for taking time to review my work.

I am grateful to those who gave me their valuable time, skills and enthusiasm, during my years as a graduate student at Georgia Tech. Dr. Hai Ying Liu and Dr. Jou Young Choi provided me with great deal of help on computer modeling. Dr. Sandeep Agarwal and Dr. Dewey Hodges gave me valuable tips and general guidance.

I appreciate all the support from engineers and CAD designers in Istanbul Technical University, Dr. Temel Belek, and Mr. Tom Hanson. Other people who have helped me include but are not limited to Dr. Han Gil Chae, Dr. Jieun Ku, Dr. Vitali Volovoi, Pete Hart and Dr. Adeel Khalid. I would like to take this opportunity to also acknowledge the rest of my professors in Aerospace, Mathematics, ISYE and Mechanical Engineering departments. My Ph.D. studies at the School of Aerospace engineering has been an invaluable learning experience. Thanks to all of the above and the many friends, and colleagues not mentioned for their advice, assistance, and friendship.

Finally I would like to thank my parents for their love, care and continued support throughout my educational career. I appreciate their patience and prayers without which this work would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

a Aerodynamic lift constant.

β Blade flapping angle.

β_0 Blade precone angle.

β_{allow} Allowable flapping angle for maximum fatigue life.

c Blade chord length.

CA Contributing Analysis.

CAD Computer Aided Design.

CAMRAD Comprehensive Analytical Model of Rotorcraft Aerodynamics and Dynamics.

CATIA Computer Aided Three Dimensional Interactive Application.

CCD Central Composite Design.

CFD Computational Fluid Dynamics.

CG Center of Gravity.

c_l Blade lift coefficient.

CO Collaborative Optimization.

CONMIN Constrained Minimization.

DOE Design of Experiments.

DYMORE A Finite Element Based Tool for the Analysis of Nonlinear Flexible Multibody Systems.

E Elasticity modulus.

ϵ_{allow} Allowable strain for maximum fatigue life.

ϵ_x Strain in x-direction.

FEA Finite Element Analysis.

FEM Finite Element Method.

G Shear modulus.

GA Genetic Algorithms.

I	Inertia tensor.
IMF	Installation management file.
IPPD	Integrated Product and Process Development.
ITU	İstanbul Technical University.
I_x	Moment of inertia about x-axis.
L_A	Aerodynamic torsional moment.
LCC	Life Cycle Cost.
LCH	Light Commercial Helicopter.
M	Bending moment.
M_A	Aerodynamic flapping moment.
M_{allow}	Allowable flapping moment for maximum fatigue life.
MATLAB	Matrix Laboratory.
m_{blade}	Blade mass.
MDF	Major dimensions file.
MDO	Multidisciplinary Design Optimization.
m_{tip}	Blade tip mass.
N	Number of load cycles for fatigue testing.
NACA	National Advisory Committee of Aeronautics.
NCAT	National Center for Advanced Technologies.
OA	Orthogonal Array.
OEM	Original Equipment Manufacturer.
Ω	Blade operating angular velocity.
ω_0	Feathering frequency at zero rpm.
PRG	Product relations geometry.
R	Rotor radius.
R_{blade}	Blade length.
R_{flex}	Flexure length.

ρ	Air density.
RPM	Revolution Per Minute.
RSM	Response Surface Methodology.
SA	Simulated Annealing.
SEF	Syntactic Epoxy Foam.
SFD	Surface definitions file.
σ_{allow}	Allowable stress for maximum fatigue life.
θ	Blade pitch angle.
θ_{tw}	Blade linear twist angle.
VABS	Variational Asymptotic Beam Sections.
V_{gust}	Gust speed.
ζ	Blade forward sweep angle.

SUMMARY

Many government agencies and corporations around the world have found the unique capabilities of rotorcraft indispensable. Incorporating such capabilities into rotorcraft design poses extra challenges because it is a complicated multidisciplinary process. The concept of applying several disciplines to the design and optimization processes may not be new, but it does not currently seem to be widely accepted in industry. The reason for this might be the lack of well-known tools for realizing a complete multidisciplinary design and analysis of a product.

This study aims to propose a method that enables engineers in some design disciplines to perform a fairly detailed analysis and optimization of a design using commercially available software as well as codes developed at Georgia Tech. The ultimate goal is when the system is set up properly, the CAD model of the design, including all subsystems, will be automatically updated as soon as a new part or assembly is added to the design; or it will be updated when an analysis and/or an optimization is performed and the geometry needs to be modified. Designers and engineers will be involved in only checking the latest design for errors or adding/removing features. Such a design process will take dramatically less time to complete; therefore, it should reduce development time and costs.

The optimization method is demonstrated on an existing helicopter rotor originally designed in the 1960's. The rotor is already an effective design with novel features. However, application of the optimization principles together with high-speed computing resulted in an even better design. The objective function to be minimized is related to the vibrations of the rotor system under gusty wind conditions.

The design parameters are all continuous variables. Optimization is performed in

a number of steps. First, the most crucial design variables of the objective function are identified. With these variables, Latin Hypercube Sampling method is used to probe the design space of several local minima and maxima. After analysis of numerous samples, an optimum configuration of the design that is more stable than that of the initial design is reached.

The above process requires several software tools: CATIA as the CAD tool, ANSYS as the FEA tool, VABS for obtaining the cross-sectional structural properties, and DYMORE for the frequency and dynamic analysis of the rotor. MATLAB codes are also employed to generate input files and read output files of DYMORE. All these tools are connected using ModelCenter.

CHAPTER I

INTRODUCTION

Complexity, cost, relatively high accident rate, and inferior performance compared to airplanes may make rotorcraft seem unnecessary in modern aviation today. However, their ability to hover for extended periods of time have made them life-savers in urban and rural emergencies, and on battle fields. Their rescue and surveillance potential render them indispensable for governments and corporations all around the world.

Rotorcraft design is, and has always been a multi-disciplinary process. According to Leishmann and Johnson, engineers have been struggling with problems related to aerodynamics, performance, and flight control for more than a hundred years, starting with the first rotorcraft built by a Frenchman named Paul Cornu in 1907 [4].

Emerging technology has enabled more efficient designs through lighter and stronger materials, less pilot workload as result of advanced flight controls, and improved safety due to powerful analysis and design tools, among other advancements. Nevertheless, incorporating several seemingly unrelated disciplines into one design in coherence is still challenging today's rotorcraft industry. There is a need for a scheme that lays out the underlying elements and conditions in order to achieve the complete design and analysis of an aircraft by means of enabling all the pertinent design disciplines that augment the design simultaneously, at the same time transferring design information from one discipline to another. The concept of applying several disciplines into the design and optimization process may not be new, but it does not currently seem to be widely accepted in the industry today. The reason for this might be the lack of well-known tools to realize a complete multi-disciplinary design and analysis

of a product.

1.1 Objective of This Research

Giesing and Barthelemy [5] provide an industrial perspective of multidisciplinary optimization (MDO) research. They lay out the issues an industrial designer may have during the setup of an MDO process, and suggest frameworks, architecture to address those issues. In that study, they point out the necessity of collaboration between industry, universities, commercial software and government labs to overcome technical challenges and needs for creating an "industrial strength MDO".

Given the current condition of aircraft industry, this research aims to propose a new method to enable engineers in some design disciplines to perform a fairly detailed analysis and optimization of the design, using commercially available software as well as computer codes developed within Georgia Institute of Technology. The particular analysis to be performed is related to the dynamic behavior of a given rotor geometry and configuration. The intention is to conduct the analysis with minimal human interaction in order to reduce the design time and human error, and to improve overall efficiency. Also, by automating the entire analysis, this study aims to identify improvements over current design methods. The ultimate goal is such that, when the system is set up properly, the CAD model of the design, including all subsystems, will be automatically updated as soon as a new part or assembly is added to the design; or it will be updated when analysis and/or an optimization is performed and the geometry needs to be modified. The designers and engineers will only be involved for checking the latest design for errors, or adding/removing features. The entire CAD model is created in such a way that it can easily be updated automatically even when a relatively insignificant part in a sub-assembly is modified by one of the analyses in any design discipline. Such a design process should take dramatically less time to finish; therefore it would cause great reductions in development time and costs. Cost

reduction as a possible outcome of this research might attract industrial entities to adapt the techniques discussed in this thesis.

Design automation, reducing human interaction, incorporating a parametric CAD tool, integrating a particular analysis into the overall design loop, and in turn, cost reduction by means of improved efficiency are the goals of this study, that will contribute to the state-of-the-art in rotor design. There have been many studies about rotor analyses and design optimizations performed over the years, but in addition to the dynamic analyses, this study incorporates a detailed CAD model that is modified instantly as the design is evolved. In addition, the analysis tools are selected and data transfer between them is set up in order for this analysis to be seamlessly integrated to the rest of the entire rotorcraft design process. The high level design and analysis encompassing other disciplines of a rotorcraft design, together with aircraft cost calculations, has been performed before both in industry [2] and in Georgia Institute of Technology [92]. However in some of those studies the depth of the individual analyses, such as rotor vibration minimization, were not adequate for preliminary design purposes. Also, none of those studies include a detailed CAD model in the design loop. This research aims to provide specific analysis of sufficient depth, as well as capability to be connected to the other disciplinary analyses.

In the following chapter, the situation of aircraft design industry is presented briefly, and then relevant publications on design methods and on vibration analysis rotorcraft are reviewed. The subsequent chapter discusses the proposed method on how to set up the software essential for a rotor vibration analysis. Next, the type and details of the helicopter rotor under consideration is presented. An optimization study is performed on the current rotor design. After that, findings on the optimization process are explained and future plans are discussed.

CHAPTER II

LITERATURE REVIEW

This chapter is intended for an overview of the publications regarding the subjects discussed in this text. The first part of this chapter focuses on traditional design principles and their possible shortcomings. The second part discusses basic theory and advantages of modern design approach. The third part outlines design optimization related studies, specifically vibration reduction of hingeless rotor blades.

2.1 Traditional Design Practice in Industry

Design department of an old fashioned corporation works in a way in contrast to what a company should implement to achieve a successful rotorcraft design. In a full-fledged rotor design process, analysis of many disciplines, including but not limited to, aerodynamics, propulsion, transmission, controls, structures, rotor dynamics, noise, economics, weight and balance will have to be carried out in a coherent manner in order to obtain a competitive product. In case of military applications, weapon systems, armament, and stealth issues will also be considered. An old fashioned company would cope with this task by having its disciplinary experts perform analysis and design on one discipline at a time, and conveying the results to the next discipline in line without having any feedback in the process.

Mavris et al. suggest that since aerodynamics and propulsion are the critical disciplines to achieve a fuel balance and vehicle performance, they are emphasized and the greatest level of effort is expended in these areas [6, 7, 8, 9, 10]. As the system design moves into the preliminary design phase and the initial configuration is frozen, hardware design considerations begin to dominate and the structures disciplines begin to play a more dominant role. Later in the design phase, the controls discipline

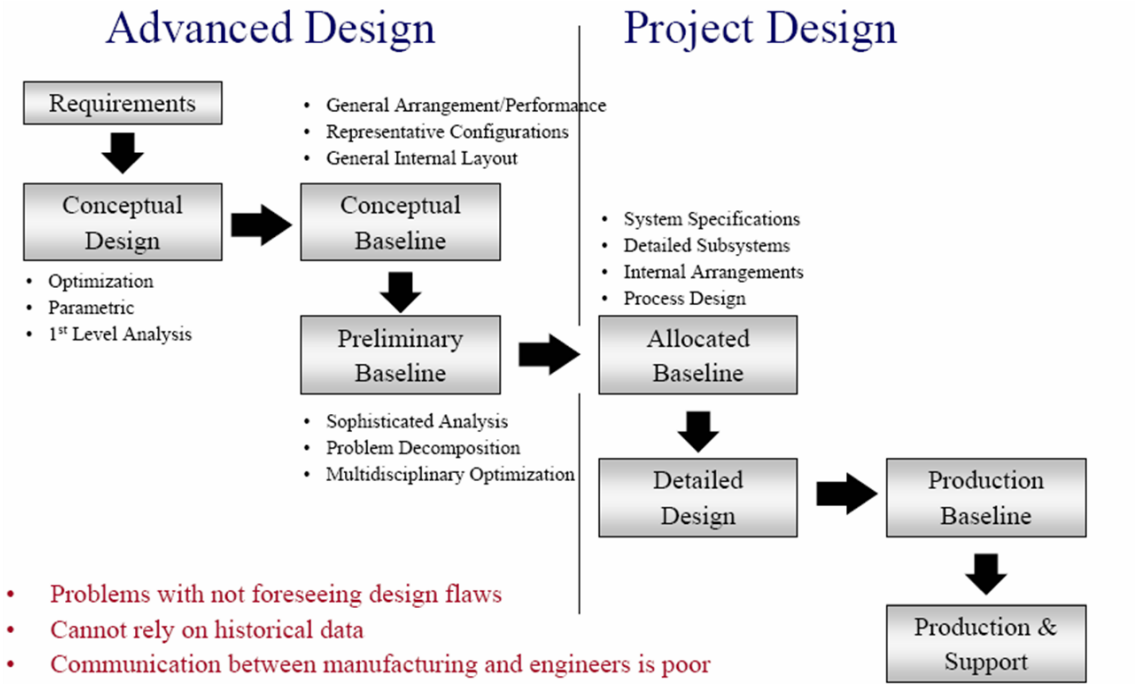


Figure 1: Hierarchical Steps in traditional Design

plays an increasing role as flight dynamics and handling quality improvements usually are necessary to achieve an acceptable flight-worthy system. Also, transition to production places a much bigger emphasis on manufacturing, cost, and to some extent supportability. The obvious problem with this traditional approach is the short conceptual design phase with an unequal distribution of disciplines, which does not allow use of design freedom to improve quality and integrate disciplines for optimization. The traditional design is also hierarchical in nature as shown in Figure 1[11].

Traditionally, for rotorcraft and most other aerospace systems, design synthesis and optimization of the overall conceptual system has been based on achieving a fuel balance and a minimum weight configuration through parametric variation of a few critical parameters such as disc loading, rotor radius, etc. This aerospace approach to design synthesis is implemented by Bates and Schrage [7], and also US army [8], and is illustrated in Figure 2 [7].

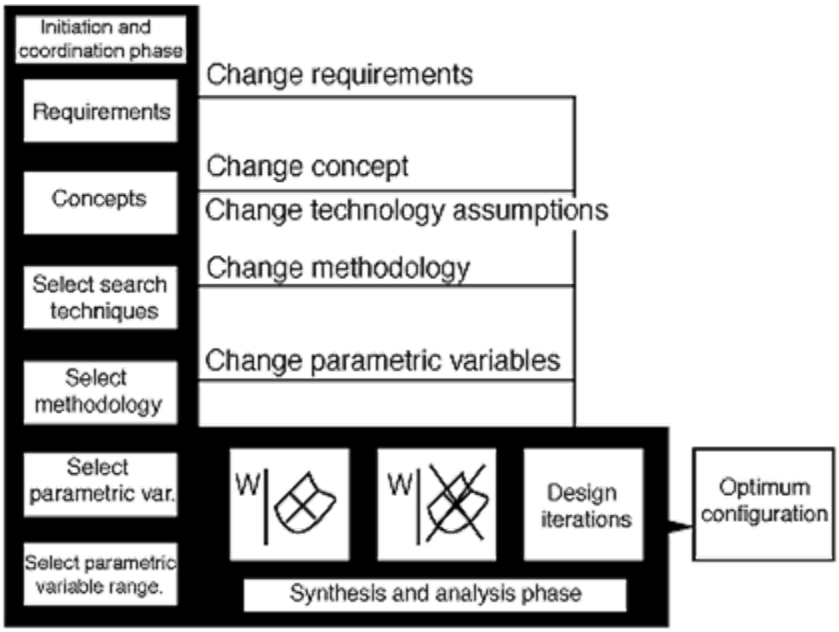


Figure 2: Present approach to Aerospace Design Synthesis

2.2 Modern Aircraft Design Processes

Safety, performance and environmental requirements of today's standards render mainstream aircraft extremely complicated systems to design and manufacture. The new Boeing 787 passenger aircraft, for instance, has approximately two million parts. There is clearly need for a comprehensive methodology to cope with design, analysis and manufacturing so many pieces, as well as managing interactions between them to achieve a synchronized system.

The rotorcraft, possibly being more detailed than a comparable fixed-wing aircraft, involves simultaneous expert attention from numerous branches of aerospace and mechanical engineering on the same sub-systems. As shown in Figure 3 [12], there are many problems unique to rotorcraft, particularly the rotor systems. Experts of disciplines such as dynamics, aerodynamics, structures and acoustics should work together to overcome possible problems pertaining to the blades, flexures, hub and coupled nature of their interactions.

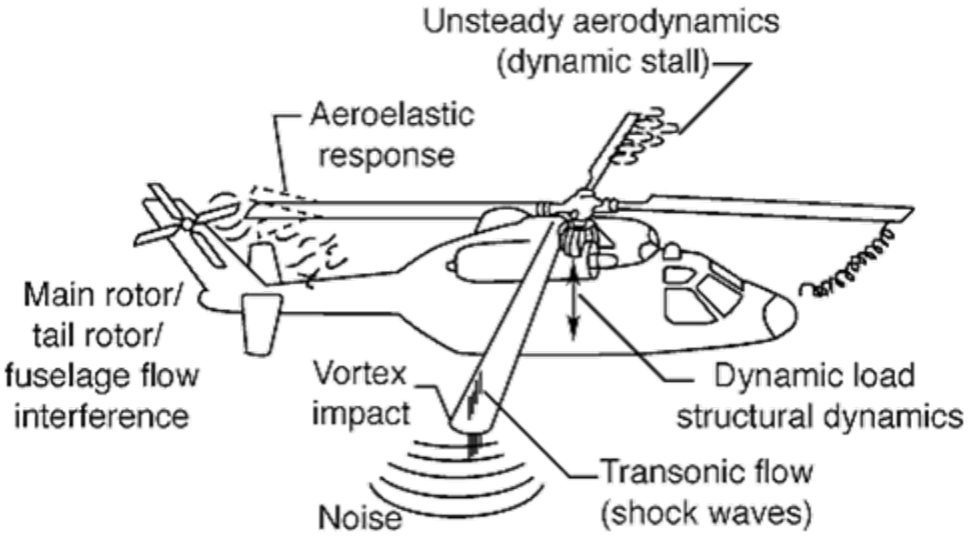


Figure 3: Helicopter as an example of a Multidisciplinary, Complex System

Necessity for an efficient design methodology is not only apparent for a superior aircraft, but also for lower costs. Cost of the aircraft implies the cost of development such as analysis and testing, manufacturing, operation and support costs and also retirement and disposal costs; rather than merely the acquisition cost of the system. All of these costs as a group are called life cycle costs (LCC) [6]. This research partially focuses on shortening time and labor required for designing the system, hence reducing the costs for development. In addition, it is desired to implement design changes as early as possible in the design process, because cost of a design change increases significantly in later design phases, as shown in Figure 4 [13]. The "IPPD Approach" in the figure stands for integrated product and process design approach. IPPD was identified in 1993 by Multi-Association Industry Affordability Task Force Executive Committee, formed by National Center for Advanced Technologies (NCAT). It was defined by the committee as a management strategy for incorporating systematic and concurrent application of all relevant design disciplines throughout the system's lifecycle [14].

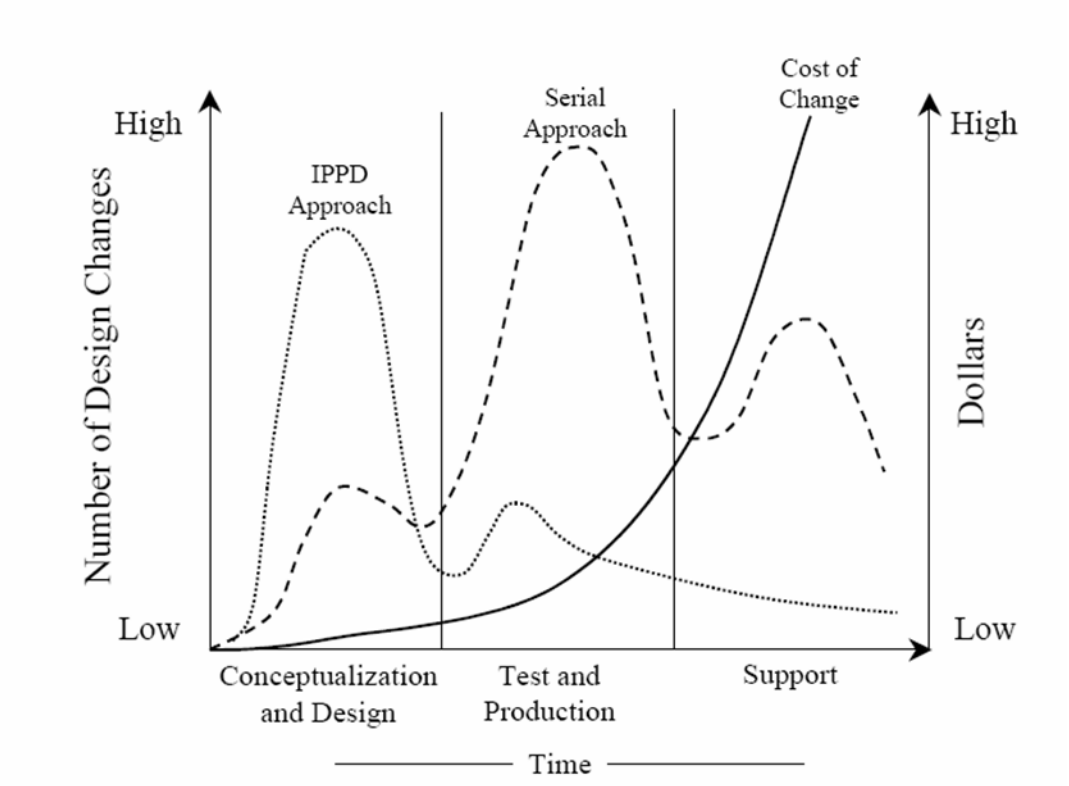


Figure 4: Comparison between serial and IPPD design approaches

Serial approach is the traditional way of design, that is, design of the system is handled by one discipline at a time. It is clear from the figure that since cost of change increases as time progresses, and most design changes occur later for the case of serial approach, IPPD approach is more cost-effective than serial approach.

Relation of time and design is not limited to number of changes and cost of change. Figure 5 [15] compares today's design process with desired future design process in terms of knowledge about design, design freedom and cost committed. It may be necessary to point out the difference between the "Cost" curves in Figure 4 and Figure 5. The one in Figure 4 represents the cost of one design change. It means the cost of changing the design increases exponentially as time progresses. On the other hand, the cost curves in Figure 5 represent the cumulative cost committed to the design process, not just an individual change. In today's design process, most of the total cost of design is committed very early in the design process, freezing most design features at concept and preliminary design stages. A traditional design company would dedicate its resources of analysis and manufacturing very early in the design. This approach does not leave adequate design freedom in subsequent stages to make improvements. Decrease in design freedom for today's and future design processes is also depicted in Figure 5. It would be more effective to have as much design freedom as possible, with minimum cost committed at the early stages. Moreover, the knowledge acquired about the design should be increased early, as well as other improvements, in order to make more informed decisions before committing costs. However, it may not be possible to obtain knowledge in all relevant disciplines in equal amounts in a traditional design setup, as shown in Figure 6 [12]; the amount of information available in the early phases of design is scattered and may be more limited in some disciplines than others. This unequal distribution of disciplines does not allow the use of design freedom to improve quality and integrate disciplines for optimization. IPPD approach may improve the situation as shown in Figure 7 [12].

The main difference between Figure 6 and Figure 7 is that with IPPD approach, the conceptual designer has more time to capture knowledge and use more design freedom. The detailed design time is reduced by up to one third based on the use of more upfront design knowledge, and a more evenly distributed effort of disciplines is provided in the conceptual and preliminary design phases.

IPPD methods help designers starting with the conceptual design stage, where there is great design freedom and almost infinite number of concepts to explore. In the preliminary design stage, however, the design space gets much narrower than that of the conceptual design stage. Moreover, the evaluation of each concept requires more complex analyses. Therefore it is necessary to organize the analysis flow in a systematic way. An IPPD framework for preliminary analysis has been developed by Schrage and modified by Chae et al. [1] [3] for rotorcraft design, where design and analysis tools are systematically arranged and incorporated into rotorcraft preliminary design stage. This framework is depicted in Figure 8. The rotor dynamics analysis in this framework utilizes the method presented in this thesis.

2.3 Design Optimization for Vibration Reduction in Rotor Blades

The rotor of a helicopter is undoubtedly its most important component. It provides all of the lift, forward, backward and sideward thrust. Analysis of the rotor usually involves multiple disciplines tightly integrated together. Rotor blades, for instance, are modeled as slender, flexible beams in most mathematical models. Even under regular operating stresses, they may exhibit large deformations in bending and torsion such that they may not be analyzed using linear beam theories. These moderately large deformations raise concerns on blade and hub stresses as well as fatigue life of the system. In addition, since blades are aerodynamic surfaces, their deformation influences aerodynamic loading, because deformed blades have different angle of attack at each crossection. This results in coupling between blade aerodynamics and

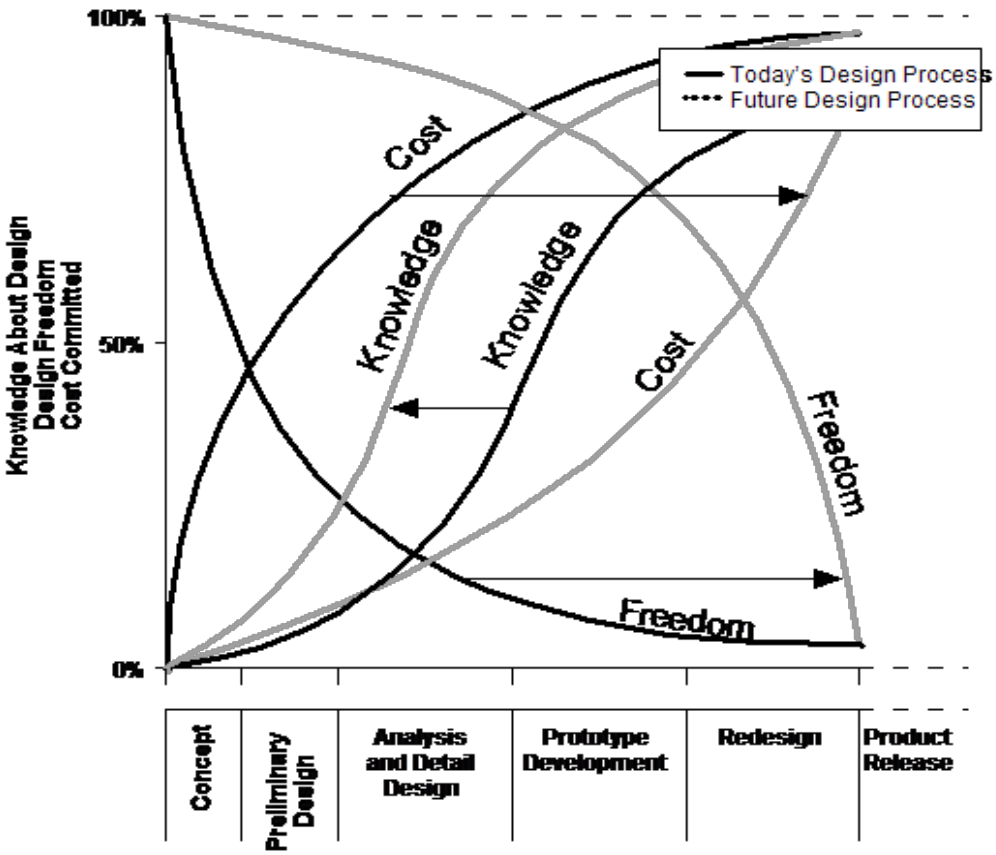


Figure 5: Design Freedom, Knowledge and Cost relationship

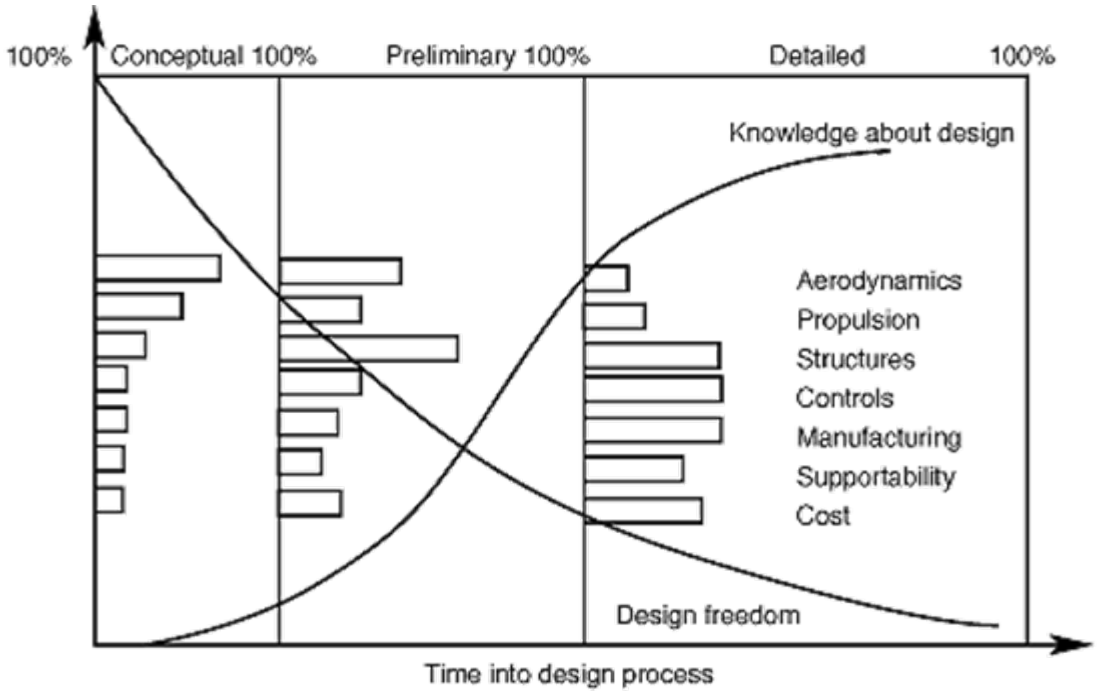


Figure 6: Traditional approach to product development

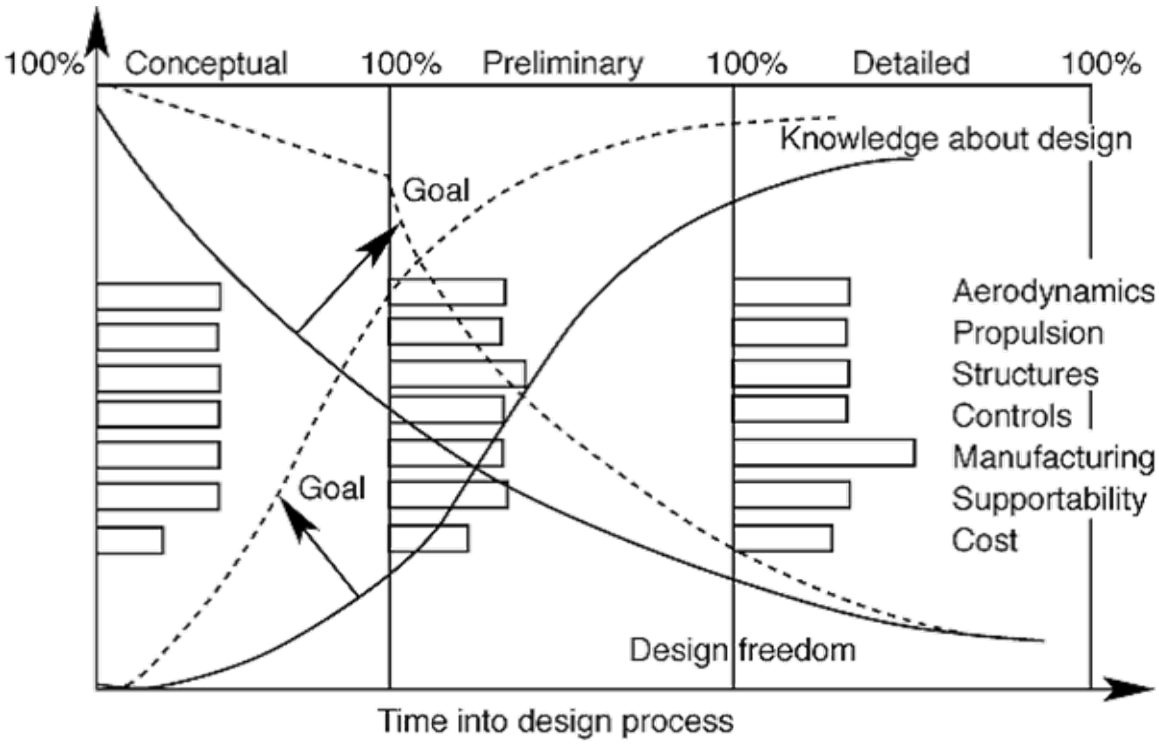


Figure 7: Design process reorganized to gain information earlier and to retain design freedom for longer

structural dynamics of the rotor.

The type of rotor considered for analysis in this text is a bearingless soft-in-plane rotor. Details of the particular rotor system to be analyzed are provided in Chapter 4. The hingeless rotor blade was mathematically modeled by Ormiston and Hodges [18] in early 1970s, with intentions of structural analysis. Optimization was not a major objective until the early 1980s [19], although some researchers like Bielawa [20] considered minimum weight design problem for a helicopter rotor in hover. Among the first publications on this subject were by Bennett [21], Friedmann and Shantakumaran [22], Davis [23], and Peters et al. [24].

Friedmann and Tong [25] derived the equations of motion of a torsionally rigid rotor blade hinged at a distance from the hub. They then investigated its aeroelastic stability for the case of coupled non-linear flap-lag motion. Friedmann and Silverthorn [26] derived equations for large amplitude coupled flap-lag motion of a hingeless elastic

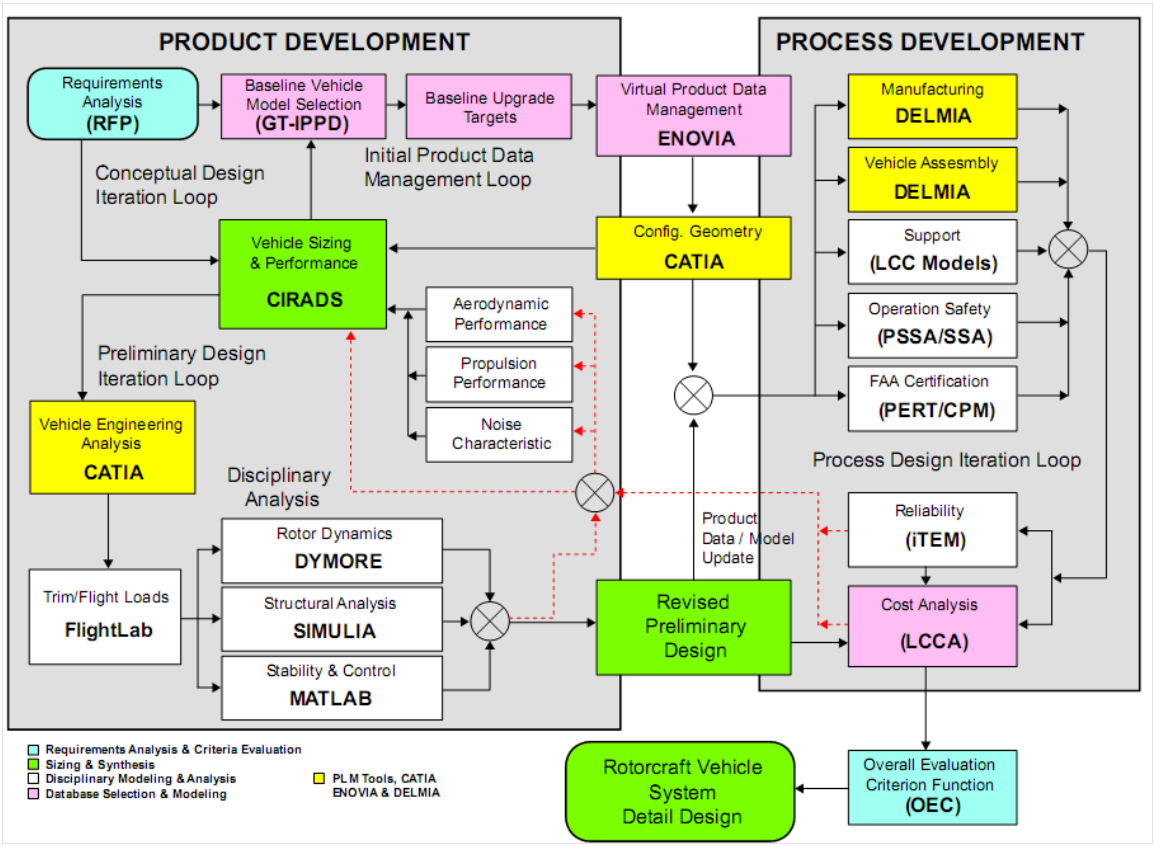


Figure 8: Georgia Tech IPPD framework for rotorcraft preliminary design

helicopter blade in forward flight. Friedmann [27] in 1984 conducted a study and stated that structural optimization offers substantial benefits in the area of helicopter rotor and fuselage design. The means of achieving a structurally optimum design is redistribution of mass and stiffness throughout the rotors and fuselage.

There are several vibration optimization studies on rotors which use either analytical or semi-analytical methods to calculate sensitivity derivatives. Although analytical approach is more exact, there are numerous studies where the alternative, semi-analytical approach is employed. In semi-analytical approach, some or all of the design derivatives are calculated using finite difference method, rather than deriving them analytically, and the resulting approximation is included in eigenproblem calculations [28]. This scheme is necessary when explicit analytical expressions for the objective function and constrains are not available. A drawback of this method is that a degree of inaccuracy is introduced due to step size. Also, computational requirements are elevated, since the complete design problem needs to be solved for every change in the design variable. Another negative aspect is that the finite difference approach does not provide insight on the significance of various parameters affecting the sensitivity derivatives [39].

Lim and Chopra [29] investigated structural optimization of a hingeless four-bladed soft-in-plane rotor to reduce oscillatory hub loads while maintaining aeroelastic stability in forward flight. Their design variables included chordwise location of center of gravity, spanwise distribution of nonstructural mass, and blade bending stiffnesses, for flap, lag and torsion. They used a direct analytical approach to derive the sensitivity derivatives of the blade response, hub loads and eigenvalues with respect to the design variables. During the optimization process, they used finite element method for aeroelastic analysis of the rotor. They utilized method of feasible directions and the computer code CONMIN to conduct optimization. They concluded that an optimum design has unchanged lag bending stiffness, higher flap bending stiffness

around mid-span and low torsional stiffness near blade root. They calculated derivatives of hub loads using a direct analytical approach, more specifically, using chain rule differentiation, in other words, partial derivatives. This technique appeared to be more efficient and reliable than finite difference approach, since numerous derivatives are evaluated analytically. Celi and Friedmann [31] performed structural optimization of a simplified soft-in-plane hingeless rotor blade to minimize the n per rev vertical hub shears in forward flight, subject to frequency constraints, aeroelastic stability constraints in hover and autorotation constraint. They used feasible direction code CONMIN for solving the optimization problem. In order to reduce computing time, they calculated gradient of the Hessian by taking differences of their current and previous values of their objective function they considered in their iterations, rather than using small finite difference steps.

Adelman and Mantay [32] presented applicability of integrated multidisciplinary design optimization into systematic analytical rotor design procedures. They explored the possibility of using aerodynamics, dynamics, structures, and acoustics simultaneously during the design instead of a sequential approach.

Chattopadhyay et al. [33] managed to minimize blade weight and 4 per rev vertical hub shear for a rotor blade in forward flight, placing constraints on first four elastic coupled natural frequencies, blade autorotational inertia, and centrifugal stress. They used an integrated aerodynamic load / dynamic optimization procedure as their multi-objective optimization method. They used CAMRAD for aeroelastic analyses together with CONMIN for optimization of the analytically modeled simplified blade. Chattopadhyay and Chiu [34] extended that work in 1992 by including additional design variables such as spanwise distributions of blade bending stiffnesses, torsional stiffness, nonstructural mass, chord, radius of gyration and blade taper ratio, and adding constraints on 3/rev radial shear, 3/rev flapping and torsional moments, 4/rev lagging moment, blade natural frequencies, weight, autorotational inertia, centrifugal

stress and rotor thrust. Furthermore, they enhanced the objective function for 4/rev vertical and 3/rev in-plane shears by including more components of the vibratory hub loads.

Davis and Weller [35] performed design optimization using several optimization algorithms on a soft-in-plane bearingless rotor. They concluded that a modified method of feasible direction and sequential quadratic programming were the two most efficient optimization algorithms among seven methods they utilized, including indirect penalty methods, approximate methods and direct methods. They have also found out that frequency placement criteria alone in the optimization constraining were inadequate for minimizing vibration. They have conducted experimental tests to verify their results.

Rotor design optimization may also be useful for improving handling qualities. Celi [36] performed an optimization study in which the torsional stiffness of a hingeless rotor blade is set to stabilize the phugoid oscillation of the rotorcraft by increasing the stabilizing effect of the rotor.

Lim and Chopra [37] carried out structural optimization on a four blade soft-in-plane hingeless rotor to minimize oscillatory hub loads in forward flight subject to aeroelastic stability, frequency placement and autorotational inertia constraints. They performed analytical sensitivity analysis, but sliced the blade into five elements for optimization with CONMIN. They achieved 60-90% reduction in all hub load components for a generic blade. Sensitivity analyses were performed via detailed mathematical derivation for aeroelastic stability and response characteristics of a rotor blade in both hover in forward flight conditions by Vankatesan et al [39], but the blade in this case was simplified to a single cell laminated rectangular box beam.

Walsh et al. [38] combined performance and dynamic analyses with a general purpose optimizer for minimizing the power required in hover, forward flight and maneuver, and also minimizing vibratory hub shear. Their design variables included

pre-twist, taper initiation, taper ratio, root chord, blade stiffnesses, tuning masses, and tuning mass locations. Aerodynamic constraints consisted of limits on power required in hover, forward flight and maneuver; airfoil section stall; drag divergence Mach number; minimum tip chord; and trim. Dynamic constraints were on frequencies, minimum autorotational inertia, and maximum blade weight. They ran the procedure for two cases. In the first case the objective function involved power required in hover, forward flight, and maneuver and dynamics. The second case involved only hover power and dynamics. They compared the designs from the integrated procedure with designs from a sequential optimization approach in which the blade is first optimized for performance and then for dynamics. In both cases, they concluded that integrated approach was superior.

Fulton and Hodges [40, 41] analyzed stability of isolated hingeless composite rotor blades in hover, using finite element method. They examined the three dimensional blade by formulating its crossection in two dimensions as a 6x6 stiffness and inertia matrices, together with a beam model, which is one dimensional. The formulations incorporate extension-twist and bending-twist coupling within the blades, which, in turn, account for composite material effects. This approach in blade modeling is employed in this thesis as well, with the aid of a specific in-house software code, namely VABS. More information about VABS is presented in Appendix.

Pritchard et al. [42] used tuning masses to reduce vibratory hub loads by systematically placing masses along blade span without adding a large weight penalty. Design variables were the tuning masses and their spanwise position. They limited natural frequencies of the blade with additional behavior constraints. They approximated objective function and behavior constraints using Taylor series.

Barwey and Peters [43] utilized integrated optimization of a soft-in-plane hingeless rotor by combining optimization with dynamics. Their composite blade model had realistic crossections, rather than box beam configuration. They analyzed the blade

crossections using a 2D finite element code. Their results indicated that the optimization process was extremely sensitive to the initial configuration, as well as the formulation of frequency constraints. They investigated the effectiveness of simpler but more numerous frequency constraints compared to complicated but fewer ones while heading for the same goal.

Vankatesan, Friedmann, and Yuan [44] formulated the sensitivity derivatives for the structural dynamic, aeroelastic stability and response characteristics of a composite rotor blade having single cell box-beam configuration in hover and forward flight, using semi-analytical approach. They discovered that higher flap and lag mode natural frequencies are more sensitive to variations in composite ply orientations than lower mode natural frequencies. They also found out that in forward flight, vibratory hub loads present a 10%-30% variation due to changing composite ply orientation angle. After integrating aerodynamics with dynamics in Ref. [38], Walsh et al. [46, 47] added structural optimization into the integrated method using multilevel decomposition. They decomposed the optimizer into two levels. They set a linear combination of performance and dynamic measures as objective function for upper level. The design variables included pretwist, point of taper initiation, taper ratio, root chord, blade stiffnesses, tuning masses, and tuning mass locations. The constraints consisted of limits on power required in hover, forward flight, and maneuver, airfoil drag, minimum tip chord, trim, blade natural frequencies, autorotational inertia, blade weight, and average strains. The lower level optimization was necessary to confirm that the blade retained structural integrity while providing the stiffness required by the upper level. The design variables here were the box beam wall thicknesses and several lumped areas that were analogous to longitudinal stringers in a wing box cross section. The lower level objective function was a measure of the difference between the upper level stiffnesses and the stiffnesses computed from the wall thicknesses and lumped areas. Lower level constraints were on the Von Mises stress at the box corners for

multiple load cases generated by several flight conditions, limits on wall thicknesses for thin wall theory, and other dimensional considerations. Henderson, Walsh and Young [48] reformulated the lower level optimization in Walsh et al.'s work using response surface method. They replaced the objective function of the lower level problem by a quadratic Taylor series in terms of upper design variables. Ganguli and Chopra [49] carried out calculation of sensitivity derivatives of blade loads and aeroelastic stability of a composite, four-bladed, soft-in-plane, hingeless rotor in forward flight, as an integral part of an aeroelastic analysis, using an analytical approach. They modeled the spar of the blade as a rectangular box-beam. The design variables were the ply angles of the laminated spar walls. They studied the influence of ply angles on the blade elastic stiffness, vibratory hub loads, and aeroelastic stability. They linked the aeroelastic and sensitivity analysis with an automated optimization algorithm. The objective function was a combination of all six vibratory hub loads and the constraints were imposed on frequency placement and aeroelastic stability in forward flight. Starting from an initially infeasible design they managed to increase the lag damping over 200%. Their results showed that there is a significant effect of lag bending-torsion coupling in stabilizing lag mode.

Ganguli and Chopra [51] carried out aeroelastic optimization of a four-bladed, soft-in-plane hingeless rotor consisting of a two-cell composite box-beam spar. They considered ply angles of the box-beam walls, whereas the objective functions were vibratory hub loads and vibratory blade bending moments. Aeroelastic stability and blade rotating frequencies were constrained. They performed optimization on three objective functions. One was a combination of hub loads and moments, another was a measure of bending and torsional moment, since that is the source of dynamic stresses for a hingeless rotor. The third one was a combination of the two. They generated results at a forward speed of μ=0.3. They concluded that optimum rotor design with minimum vibration transmitted to the fuselage has a detrimental effect

on blade fatigue life. In a related study, Ganguli and Chopra [52] approached the same problem by calculating sensitivity derivatives by using semi-analytical approach. They conducted the optimization using the optimizer CONMIN. They managed to reduce the objective function up to 33%.

Rotor vibration analysis using FEM usually requires detailed computer modeling of the rotor and requires extensive computing time for complete analysis. High cost of numerical analysis initiated a set of techniques known as approximation concepts. These techniques include generation of explicit approximations to objective function and behavior constraints, design variable linking, and temporary constraint deletion [19]. One method of approximating the functions is using response surfaces. In 2002, Ganguli [56] suggested using response surface methods for designing optimum rotors for low vibration. He imposed constraints on aeroelastic stability, and he imposed move limits on blade elastic stiffness design variables. He constructed response surface approximations for the objective function. Defining the objective function as vibratory hub loads, he found that second order polynomial response surfaces constructed using the central composite design sufficiently represents the aeroelastic model (within 1-6% accuracy) in the vicinity of the baseline design. He achieved about 30% reduction in the objective function using this method. He also stated that he was able to decouple aeroelastic analysis from the optimization computation. This decoupling means the computationally time-consuming aeroelastic analysis does not have to be repeated in optimization iterations.

Until recently, the predictive capacity of even the most sophisticated helicopter aeroelastic analysis codes were inadequate, as evidenced by Hansford and Vorwald [53], where hub load predictions from several codes are compared to flight test data. This may be essentially due to complexity of physical modeling. Even within one discipline, aerodynamics for instance, as the blade completes one revolution, it encounters transonic flow, reverse flow, stall, and unsteady effects including dynamic

stall. Moreover, large azimuthal variations in lift result from changes in dynamic pressure and angle of attack. The trailed and shed vortices leaving the blade result in a nonuniform wake. Other disciplines have their unique issues, complicating the problem further.

Optimization studies have also been performed on aeromechanical stability issues, namely, ground and air resonances. Obtaining sensitivity derivatives for aeromechanical problems was considered by Spence and Celi [45], and Shih, Spence and Celi [50] through using semi-analytical approach. Gandhi and Hathaway [54] used aeroelastic couplings to eliminate auxiliary lag dampers in the design of aeromechanically stable helicopters with soft-in-plane rotors. For optimization, they used pitch-flap coupling and pitch-lag coupling as design variables. In a later study, Hathaway and Gandhi [55] included blade flap and lag stiffnesses as design variables, in addition to the ones in their previous study. They imposed constraints to prevent excessive changes in the rotor frequencies. They found out through numerical results for a soft-in-plane rotor, that it is possible to eliminate lag dampers from a rotor using aeroelastic couplings and changing blade stiffness and damping properties, along with leading gear stiffness and damping properties. In addition, considering pitch-flap and pitch-lag couplings concurrently as design variables during the optimization proved to be superior in contrast to the sequential approach where the blade stiffness and frequency targets were set before incorporating the aeroelastic couplings.

More publications can be found based on helicopter rotor vibration reduction. Friedmann [58] and Celi [19] provide reviews of helicopter optimization research in the nineties and eighties. Ganguli [57] published a survey of researches focusing on the developments in the late nineties and early 2000's.

Response surface approximations among other novel optimization techniques continued to be applied to rotor optimization. Murugan and Ganguli [59] used a two-level

approach for optimization of a four bladed soft-in -plane hingeless rotor. At the upper level, their objectives were to reduce the 4 / revolution oscillatory hub loads and increase the lag mode damping. They approximated the objective function and constraints using response surfaces in order to find the optimal blade mass and stiffness properties for vibration reduction and stability enhancement. They expressed the approximate problem in terms of quadratic response surfaces and solved it using a gradient-based method. They reduced vibration in forward flight about 15%. At the lower level, they designed a composite box beam to match the upper level beam blade stiffness and mass using a genetic algorithm, which permits choosing discrete materials and ply angles.

Glaz, Friedmann, and Liu [60] investigated effectiveness and accuracies of kriging, radial basis function interpolation, and polynomial regression surrogates. In addition, they performed an optimization study by creating surrogates for vibratory hub shears and moments. They used those surrogates to generate an objective function for optimization of rotor blade vibrations. Their design variables were cross-sectional dimensions of the structural member of the blade and the non-structural masses. They compared their results with a baseline blade. They found that kriging surrogates were best for approximating vibratory hub loads over the entire design space and that the surrogates could be used effectively in helicopter rotor vibration studies.

In another attempt to simplify optimization process, Bhadra and Ganguli [61] used orthogonal array-based metamodels in the aeroelastic optimization of a four-bladed, soft-in-plane hingeless rotor in forward flight (μ=0.3), in order to reduce hub loads and blade root loads with blade stiffness design variables. They created metamodels using experimental designs such as factorial designs, central composite designs (CCD), gradient-enhanced CCD, and orthogonal arrays (OA). They obtained linear, quadratic and cubic polynomial response surfaces. They concluded that face-centered CCD and OA using quadratic polynomial approximations produce acceptable accuracy.

In a recent study in Georgia Tech, Ku [62] optimized flexure and blade spar of a soft-in-plane, bearingless composite rotor for minimum weight. She used Genetic Algorithms and then gradient-based methods for reaching the optimum, while conforming to frequency placement and autorotation index constraints. She created surrogate models in order to capture general shape of otherwise complicated design space, using design of experiments (DOE) methods. She also tested response surface method (RSM) and Kriging. Khalid [92] optimized the design of a helicopter using the same rotor for minimum overall cost. In addition to dynamic analyses, he incorporated vehicle engineering, stability and control, aerodynamics, propulsion, transmission, weight and balance, noise and cost calculations on a highly integrated framework. However, although he performed all of those analyses, he did not use high fidelity techniques in most of his calculations. This thesis proposes a high fidelity approach to rotor flexure geometry optimization similar to Ku's work. In addition, this thesis attempts to establish the connection and data transfer between the analysis tools to be seamlessly fit in the framework Khalid employed in his thesis, as realized by Chae et al. [3]. In that study, Chae et al. used IPPD methodology to design and analyze the performance of the same rotor Ku and Khalid considered, but for a different mission requirement. On top of the analyses for product design such as CFD, dynamics, stability and control, they included process design tools as well to perform virtual manufacturing and analysis for cost. All the disciplinary analysis in that study were tightly linked for data transfer, using the same framework as that of Khalid's work.

Collins et al. [64] used both high and low fidelity tools for the purpose of reducing noise of a helicopter rotor, incorporating several disciplinary analyses. Their low fidelity model assumed rigid blades, prescribed wake aerodynamics, and compact blade loading for aeroacoustics. After identifying one particular design possibility among millions of others using Monte Carlo simulation, they switched to higher

fidelity CFD based tools. They optimized the design to obtain better performance characteristics and reduced noise. In his thesis, Collins [63] used similar tools and techniques to perform CFD and CSD analyses on a 4 bladed hingeless rotor. The software tool he used for linking different analysis tools is the same as the one used in this study.

In 2009, Orr and Narducci [2] developed a plan for high fidelity optimization process for rotorcraft. They used the same software tool for data transfer between disciplinary analyses as Khalid, Chae, Collins and this thesis. They established optimization objectives for several disciplines including aerodynamics, dynamics, structures, drive system, propulsion, flying qualities and noise for both rotor and fuselage, where applicable.

CHAPTER III

HYPOTHESIS AND PROPOSED METHODOLOGY

This research aims to answer the following questions:

- How would automated analysis improve current rotor design methods?
- Where in the rotor blade design process would this particular analysis fit?
- Is it possible to analyze and plot rotor dynamic behavior for a given rotor geometry and configuration without human interaction?
- How can a CAD tool be incorporated into the automated analysis loop to modify rotor geometry in real time?

These research questions are attempted to be answered by performing the following tasks:

An automated analysis is theoretically free of human error, since there is no human input during the analysis loops. Elimination of human error should improve efficiency and accuracy of the final design. Automation might also reduce the time required for the design. Performing the required tasks individually in each relevant software may take much more time than having them accomplished by integrating the same software tools. One drawback of automation, on the other hand, could be the time required for setting up the system. In order to compare the time necessary for manual and automated analysis, time for manually conducted analyses is roughly estimated based on the author's experience. The time it takes for the automated analysis performed by the computer is also recorded, and added to the automated-system setup time. Details of the estimation and comparison of the two analysis methods are discussed in Chapter 5. A graphical clarification of this strategy is given in Figure 9.

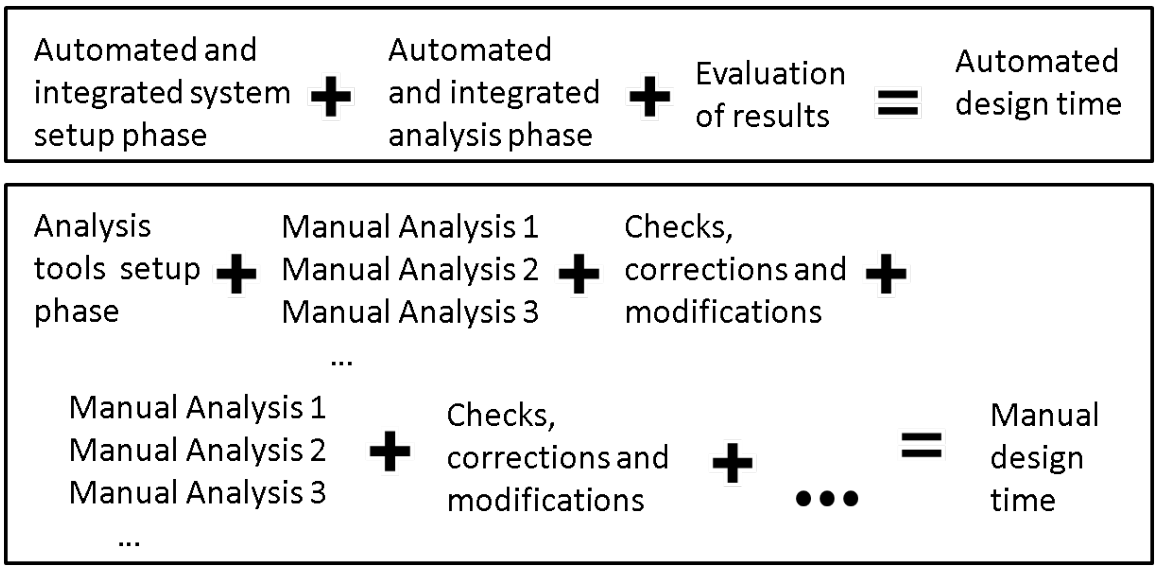


Figure 9: Comparison of design time between manual and automatic analyses

A design process is traditionally divided into three stages: conceptual design stage, preliminary design stage, and detail design stage. In the conceptual design stage, customer requirements and needs for a new product are identified and evaluated, and then several design alternatives are considered. Since there are usually hundreds of design possibilities, no detailed analyses are performed at this stage. However, calculations based on historical data, statistics, and greatly simplified analyses are possible. Engineers usually arrive at a few design alternatives to be passed on to the next stage at the end of conceptual design. During the preliminary design stage, subsystems of the product are designed and specified through extensive analyses. Interfaces between the subsystems are defined as well. The last design stage, the detail design, includes the development of the detailed designs with manufacturing processes, tolerances etc. that enable the production of a prototype.

Based on those definitions of design stages, the type of analysis performed in this study fits best into preliminary design stage. The type on analysis is fairly detailed, and it applies to a specific subsystem. The design process may also be considered as an analysis and optimization problem within the preliminary design stage, encompassing

all the design disciplines pertinent to the product. An example of an optimization setup is depicted in Figure 20. In this case, this study partially involves in a number of analyses: vehicle engineering, structural analysis and rotor dynamics. The manner in which this study takes those analyses into account is described below, and an example case is considered in the next chapter.

Minimizing human input during design is the major goal of this study. The means of eliminating human interaction in rotor analysis starts with the CAD model of the design. The geometrical information from the CAD model is combined with material properties and composite layup details. This information about the product is then combined with loading conditions for the particular analysis to be performed. At the end of the analysis, the CAD model is updated automatically according to the results. This process is explained in detail later in this section under the heading *Relational Design*.

The hypothesis of this study is as follows:

(a) A relatively complicated and multidisciplinary mechanical system such as a helicopter rotor can be designed or optimized using relatively high-fidelity analysis and modeling software.

(b) Performing analysis and optimization in a digital environment enables automation of design process. Provided the analysis environment is set up properly, an automated design process enables little or no human interaction with the computer until the design is completed.

The above hypothesis is tested by using a bearingless, soft-in-plane helicopter rotor as the complicated multidisciplinary mechanical system. The rotor system is originally designed by an experienced rotor designer, Thomas F. Hanson. It can stabilize itself during operation under arbitrary gusts without need for electronic stability control systems or manual control input. Because of this feature, Hanson calls it the "Auto-Trim rotor". The rotor has one more advantage over most other

rotors; there is no need for power assistance such as a hydraulic system for controlling the swash plate. The first natural frequencies in flapping and feathering motions of the blade are set close to the operating RPM in order to bring the resonance frequency close to control frequency. Details of the rotor system are explained in the next chapter.

The designer of the Auto-Trim rotor system, Thomas Hanson, is a retired helicopter designer. He has almost half a century of rotor design and field testing experience in Piasecki, Convertawings and Lockheed Corporation. After leaving Lockheed in 1967, he worked as a consulting engineer. He believes a helicopter should be designed by a single person, probably a mechanical engineer, who is knowledgeable in all the required disciplines. This way he/she can create connections between all disciplines in his/her mind and therefore come up with an efficient and consistent design, without any conflicts between various disciplines.

Hanson was an active aerospace engineer in the nineteen fifties and sixties, in which time computers in design were almost nonexistent in major aircraft companies. It is only natural for an engineer at that time to think of a human mind to be the only medium to incorporate all the analysis and design with full communication between disciplines at all times.

Considering the inherent complexity of rotorcraft together with the need for specialized expert input for each design discipline, it is almost impossible for an individual to possess adequate depth of knowledge about every single discipline. Especially in today's level of technological advancements, becoming an expert in any given discipline requires years of academic work and industry experience. Even if such an individual existed, it would still require others to inspect, test and verify the work of that individual. Therefore, he/she would still have to communicate with other engineers in order to inform them about his/her decisions and analysis.

Today's computer technology and available software tools manage communication between various design disciplines to a degree such that the analysis tools can transfer design data to and from each other instantly while the design process goes on. This would mimic the mind of a single design engineer in terms of data management and communications within design disciplines. Moreover, it is possible to use the vast processing power of computers to perform a design optimization. This capability is beyond human capacity because a global optimization process considers all design parameters from all design disciplines simultaneously.

3.1 Analysis Tools

It may be possible to come close to Hanson's dream of one mind doing the entire design work, using appropriate software tools for analysis of the design in several disciplines, and one software tool for setting the framework of communication between analysis software. There is such a tool for information transfer commercially available: *ModelCenter* by Phoenix Integration Inc.

ModelCenter is a software environment designed for integrating originally unrelated software packages. It enables conducting complex design exploration tasks using a wide range of supported commercial analysis software or simple command-line based executables. The design data is transferred from one program to another automatically, eliminating the need for manually converting output-input file formats of incompatible analysis tools. It can be used as an inactive environment serving as the common medium for communication of programs.

Capabilities of ModelCenter extend beyond automation and communication environment. It can perform parametric design studies, design of experiments, response surface modeling and optimization. The optimization tool uses line search methods such as method of feasible directions, conjugate gradient method, and sequential quadratic programming.

ModelCenter can control input and output parameters of CATIA, MATLAB, Excel and several other programs. It is also possible to prepare a wrapper code for ModelCenter to control command-line executable applications such as DYMORE. Those tools and applications are briefly introduced in the following paragraphs.

This research focuses on structural and dynamic aspects of rotor design, rather than others. The current rotor design is analyzed and optimized with respect to structural stability and dynamic response to external excitations. The most suitable software readily available to the author for this task is *DYMORE*. DYMORE is a finite element based tool for the analysis of nonlinear elastic multibody systems. Multibody models to be analyzed are created using a library of pre-defined elements such as rigid bodies, cables, beams, shells and joints. It is possible to define a multibody system in DYMORE composed of 1D elements configured in 3D space, and then define crossectional 2D structural and material properties to the 1D elements to perform a 3D analysis. Basic information about DYMORE is presented in the appendix. A detailed description and features are also available by the author of the code in ref. [88].

Once the dynamic and structural analyses are performed in DYMORE, it may be necessary to make changes in the design in order to conform to natural frequency constraints of the structure in vacuum or stable dynamic behavior of the system in operation. Design change usually implies a modification in the geometry of the system. The geometry of the Auto-trim rotor is modeled using a CAD tool called *CATIA*. Basic capabilities of CATIA are discussed in the appendix.

The analysis loop in this study was performed in two ways: First method is using the old-fashioned way of modeling and simulation separately in each of the analysis and modeling tools, in a relatively "manual" way. Second method is setting up a system of integrated modeling and simulation environment to obtain an optimized design through several automated iterations. The analysis steps for each case is

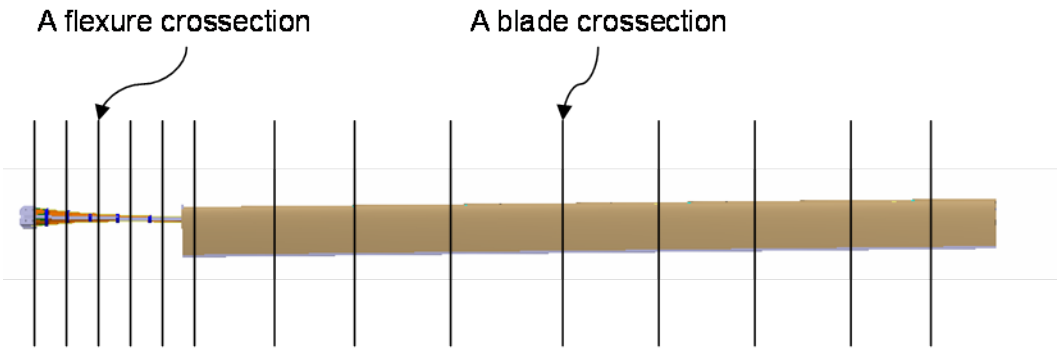


Figure 10: Crossections of the rotor

described below.

3.1.1 Old Fashioned modeling and analysis method

When the geometric features or dimensions of the design are modified, its structural properties may have been altered as well. It is necessary to extract structural data after each modification. In the case of the old-fashioned analysis, for the sake of simplicity and compatibility with in-house analysis codes, the 3D CATIA model is divided into several slices perpendicular to flexure and blade span axes, as depicted in Figure 10. Each of these 2D crossections are stored in separate CATIA files, and then imported into a structural finite element analysis (FEA) tool. Two representative 2D crossections are shown in Figures 11 and 12. The FEA tool to be utilized in this study is *ANSYS*. The reason for choosing ANSYS over other commercial programs is that it is possible to process a 2D ANSYS finite element model using an in-house (i.e. developed and used by researchers at Georgia Tech) sectional analysis tool, *VABS*.

VABS is a computer program that implements a variable asymptotic method for computing the stiffness of a heterogeneous beam at a given cross section. The variational-asymptotic method is a rigorous mathematical technique by which a 3D representation of a thin elastic body such as a beam or plate can be methodically reduced to a 1D or 2D model. VABS provides an accurate 1D model of a rotor blade for a given level of complexity. The resulting cross-sectional stiffness matrix exhibits

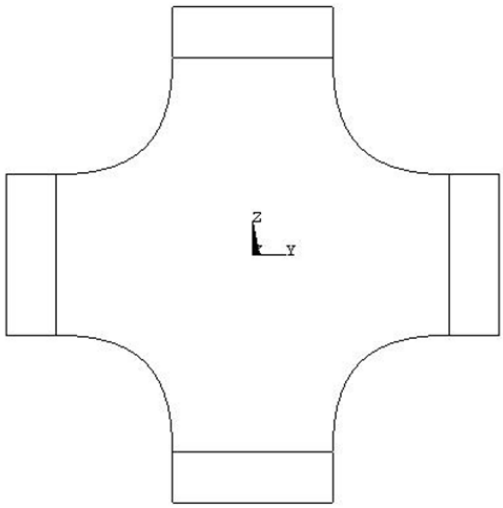


Figure 11: A flexure crossection

Figure 12: A blade crossection

full elastic coupling, allowing a designer to take advantage of composite materials when designing rotor blades.

VABS is capable of computing the full 6-by-6 stiffness matrix with all the bending, extension, shear and torsion terms, with asymptotic accuracy. The sectional stiffness can then be used in 3D flexible multi-body analysis tools such as DYMORE where elastic coupling can readily be accommodated to model the response of complete blades and rotor systems. VABS can provide an additional benefit of computing the stress distribution on the cross section if the appropriate sectional resultants (i.e. bending moments, twisting moments, transverse shear forces and axial forces) are provided. See the appendix for more information on theoretical background.

In order to use VABS to compute sectional stiffnesses, it is necessary to create a suitable 2D discretization of the crosssection that should result in a 2D computational mesh. The *VABS-ANSYS Toolset* is designed to address this mesh generation issue as well as to provide visualization of the analysis results after VABS is executed. As implied in the tool name, ANSYS is used in this tool to handle the mesh generation and visualization of the results.

Once the finite element model for each crossection is generated in ANSYS and exported as VABS input files using VABS-ANSYS Toolset, the input files are processed by the VABS executable application. The 6-by-6 stiffness matrix of each crossection is the most important result for the purposes of this study. All the stiffness matrices are entered into DYMORE input files as beam crossection properties along both the flexure and the blade models.

Examining the DYMORE output, the designer can make changes on the CATIA model, and then redo the analysis loop for the next iteration. The procedure is summarized in Figure 13.

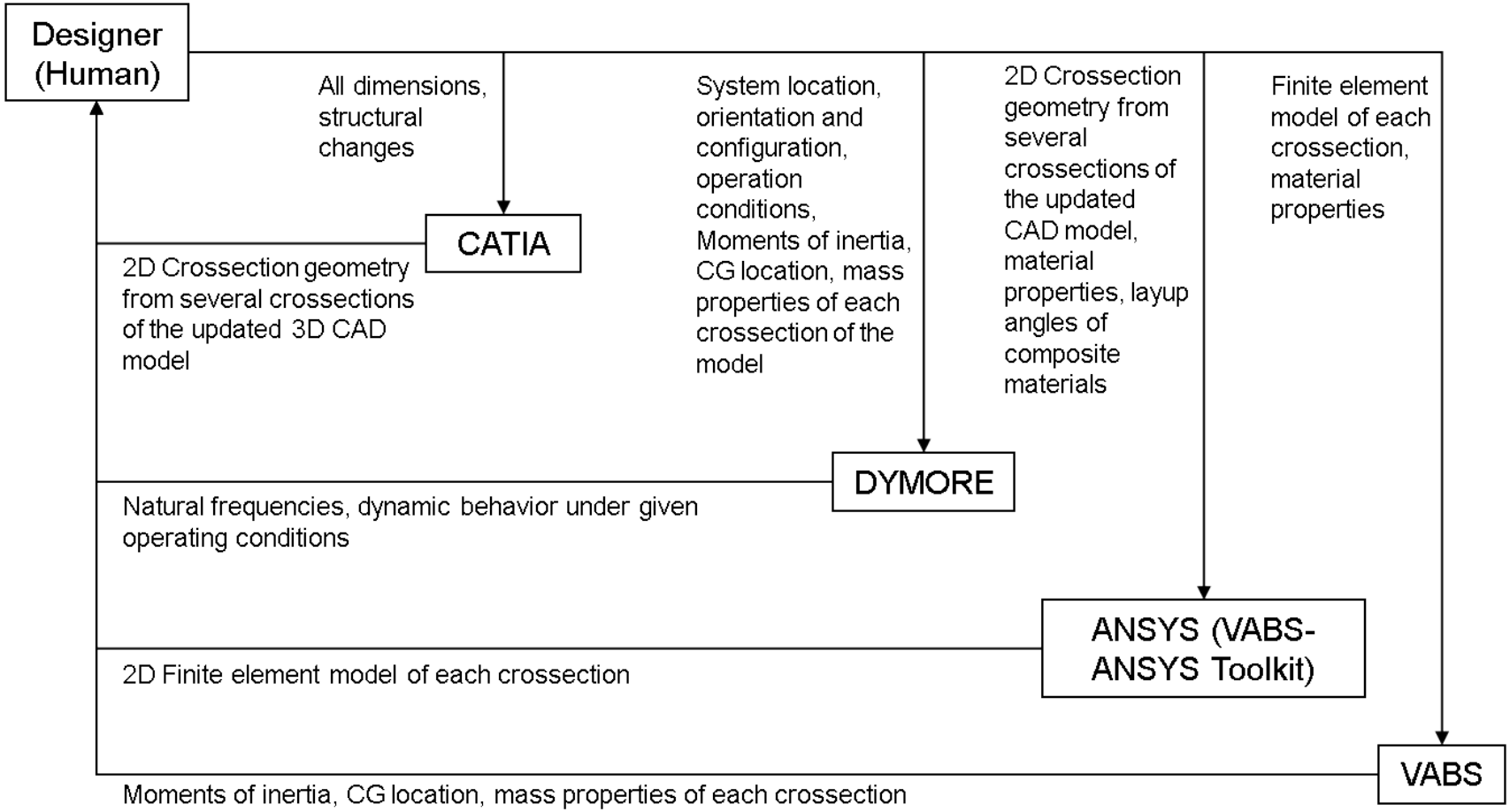


Figure 13: Initial Setup for theoretical flow of information for the manual case

3.1.2 Automated modeling and analysis method

The steps in the automated case are similar to the ones explained in the previous section, but have some differences. In this case, in addition to the tools utilized above, three more software tools are introduced. These tools are Microsoft Excel, MATLAB and ModelCenter.

Excel is used for entering the geometrical information about the model, such as dimensions and angles, and also the material properties necessary for ANSYS and VABS. In addition, some simple calculations are preformed in separate Excel sheets for preparing DYMORE input files. All this information is then transferred into CATIA, ANSYS and DYMORE through the integration software ModelCenter. Using Excel for defining the major characteristics of the model saves time such that the designer does not need to have an extensive knowledge of the complex analysis tools such as CATIA to run the analysis, since the CATIA model is fully linked to Excel. Whenever a dimension is updated in ModelCenter, it is updated in Excel as well. This update is almost immediately transferred into the CATIA model; therefore the CAD model with the latest geometry is obtained without having to make any changes in CATIA. This rationale also applies to ANSYS macros and DYMORE input code, since they too are fully integrated to the ModelCenter environment.

MATLAB is necessary for analyzing DYMORE output. The typical output of a DYMORE analysis is a text file containing the natural frequency information of the model for static analysis; and several text files containing forces, moments, displacements and rotations of each node in the model for dynamic analysis. In an automated optimization procedure, the data in the output text files have to be processed and converted into other forms to be recognized by the ModelCenter optimizer. A MATLAB code is written to open the text files generated by DYMORE, extract relevant information, and transfer it to the optimizer in a concise manner.

The data transfer between several tools explained above is depicted in Figure

14. In the figure, as data flows from one analysis tool to another, ModelCenter acts as an environment to enable communications. The "Optimizer" box represents an optimization tool for any user-defined objective function. The user may pick an optimizer available within ModelCenter, or link to third party software in the same manner as the other analysis tools.

Utilization of ModelCenter is limited to performing data transfers between various programs and processing analysis outputs for current design analysis and optimization. Once the updated input is fed into a linked analysis tool, ModelCenter waits for the output of the tool without interfering with its internal update procedure. If the analysis tool is a simple command line executable application such as VABS or DYMORE, the update procedure consists of reading the input text file and then generating an output text file. The output file or files are interpreted by ModelCenter in order to modify related design variables. The updated variables are then transferred into the next analysis as new input. However, if a particular analysis or modeling tool or is complex enough, it may require a specific procedure of modeling or coding in order to render the tool ready for model modifications. CATIA is such a tool. If a part model geometry is defined without further modifications in mind, it may be cumbersome, if not impossible to update the part dimensions and/or features depending on the progress of the global analysis. All the dimensions and geometrical features that are prone to future updates must be parameterized. Moreover, complex systems like helicopter rotors are never comprised of only one part. They are usually intricate assemblies of several detailed parts with varying sizes, shapes and materials. Shape of one part is usually dependent on the shape of its adjacent part. The CATIA model of the entire assembly must be set to define relations of each part or subassembly with each other precisely in order to avoid erroneous updates and invalid geometry. A modeling method called "relational design" is used for setting CATIA models correctly for this study. This methodology was originally developed for implementing

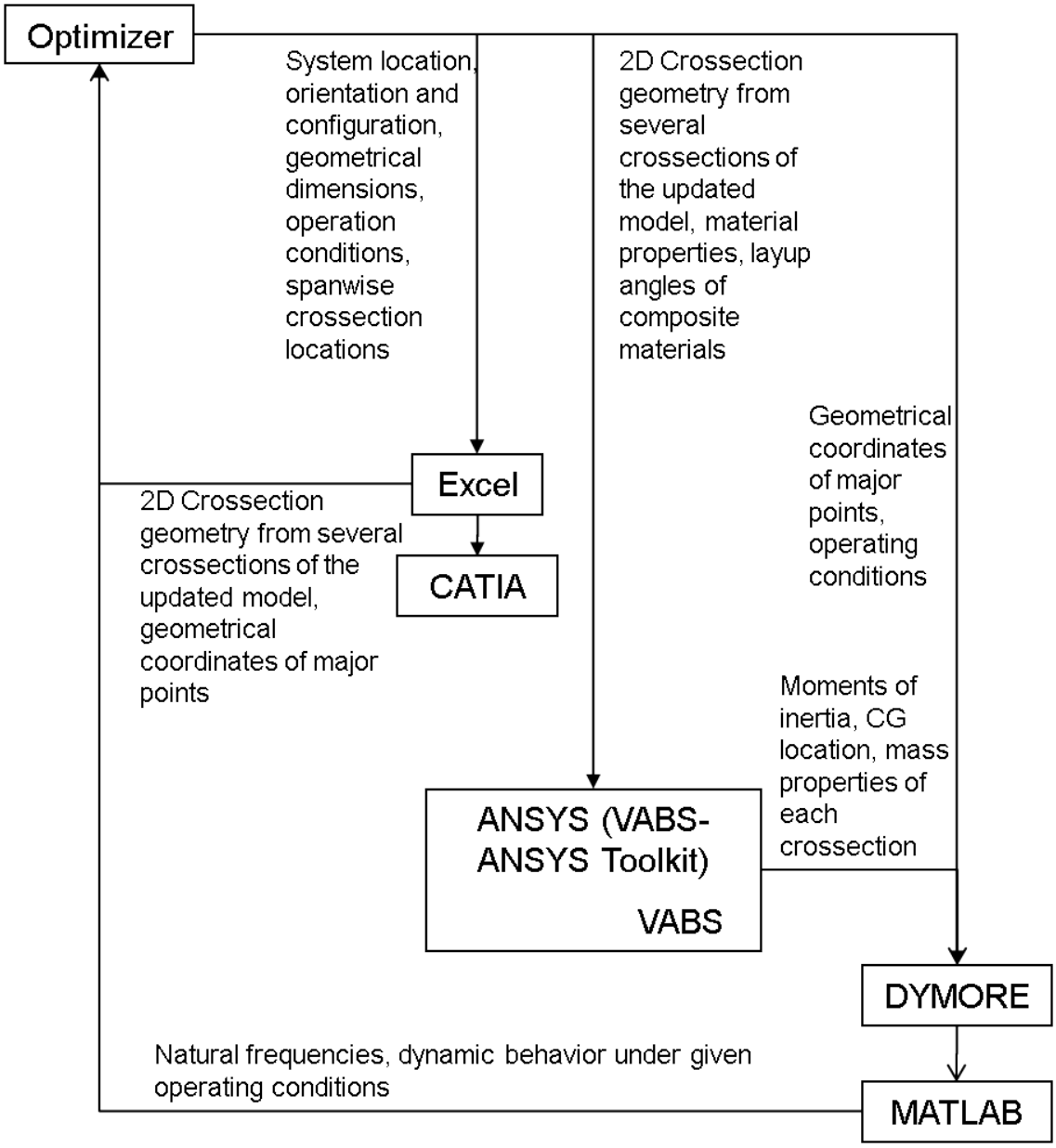


Figure 14: Initial Setup for theoretical flow of information for the automatic case

simultaneous aircraft design by separate companies located far from each other. It enables fast modification of the entire product when one company makes changes in one part. Basics of relational design are discussed in the next section.

Hanson's initial rotor design is analyzed by DYMORE and other third party tools, but the optimization of the system is performed by ModelCenter. As mentioned above, ModelCenter has numerous options for analyzing design possibilities and conducting an optimization study. Basics of optimization and optimization plan in this study are discussed in the last section of this chapter. Focus here is the interaction and data transfer between various software tools. Figure 14 illustrates the objective of this research. The process shown in this figure is explained step by step as follows:

1. Develop an Excel sheet with calculations for geometrical shapes and sizes of major components of the design. Also set the spanwise locations of the crossections of the geometry to be analyzed.

2. Create the product geometry from scratch in CATIA using relational design guidelines. Use the lengths and angles from the calculations in the Excel sheet defined in the first step. This is not an automatic process. The product has to be modeled by a designer. This step is not included in Figure 14, but the time required for this task is taken into account in Chapter 5 when comparing the old fashioned design method and the automated design methodology proposed in this study.

3. Link the CATIA model into ModelCenter environment. This is realized by using a third party "CATIA wrapper" software specifically written for this task.

4. Write an ANSYS macro to generate each flexure cross-section. The macro will also have ANSYS assign materials to particular areas of the geometry, generate a mesh and create the input files necessary for VABS. It will then run VABS to extract moments of inertias and 6x6 stiffness matrices of each 2D cross-section.

These matrices are saved as text documents by default. ModelCenter can run this macro, open the text documents created by VABS and extract the stiffness matrices. The execution of ANSYS and VABS is depicted graphically in Figure 16.

5. Setup DYMORE to be run within ModelCenter by means of a "file wrapper" to generate input files for DYMORE. When new parameter values and product structural properties are available in the ModelCenter environment, a new DYMORE model is generated by means of creating text files to be read by DYMORE executable.

6. Link DYMORE to MATLAB. MATLAB is required for processing output of DYMORE analysis. DYMORE output files are mostly text files with lists of numbers. These numbers can be presented in a more meaningful way in terms of fan plots and graphs of structural response behavior. The same MATLAB file is used for defining the objective function for the optimizer as well.

7. Link the MATLAB file generated in the last step with the ModelCenter optimizer. The last 3 steps are shown in Figure 15, isolated from the rest of the process.

8. Set up the optimizer in ModelCenter with important parameters, objective function and constraints clearly defined.

9. The system is now setup. Run the optimizer to begin the automated optimization process. It should automatically run Excel, MATLAB, DYMORE, CATIA, ANSYS and VABS several times until the optimum of the objective function is achieved.

10. When the optimization is complete, the designer has to check the result against

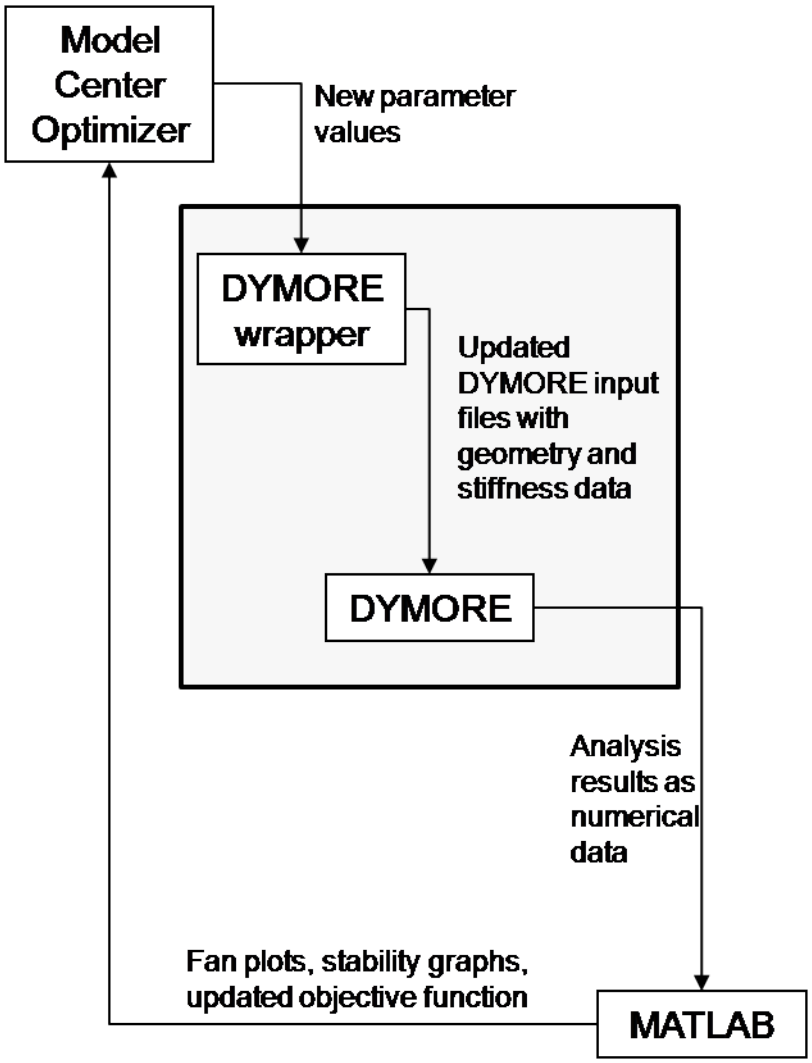


Figure 15: Interaction of ModelCenter optimizer with DYMORE

engineering sense. The process may need to be repeated with revised parameters and/or loading conditions if the optimum does not lead to a reasonable configuration. If the result is acceptable, then the designer may need to generate a report for related disciplinary experts based on his or her particular analysis. This step is not explicitly included in the proposed design methodology, but the time required for this step is estimated nevertheless.

The steps identified above define the flow and type of data transfer during the

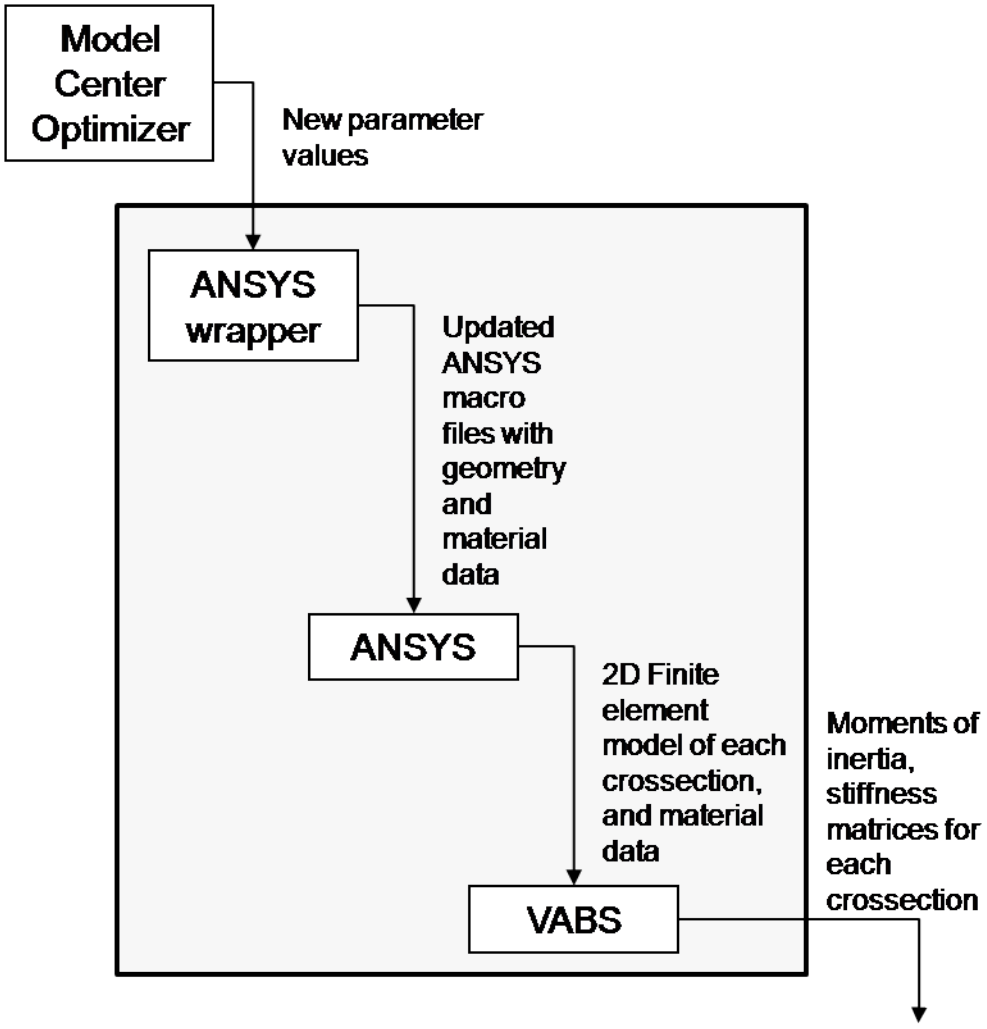


Figure 16: Interaction of ModelCenter optimizer with ANSYS and VABS

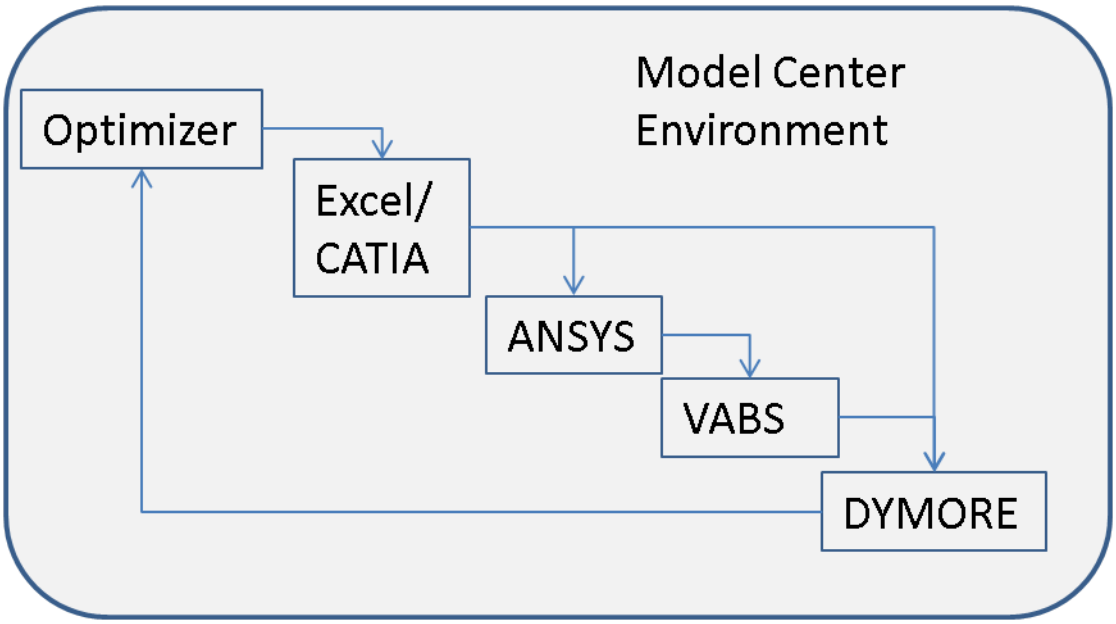


Figure 17: Simplified illustration of the optimization process

optimization procedure shown in Figure 14. A simplified diagram of the same procedure is also given in Figure 17. In this figure, MATLAB is intentionally omitted although it is an essential component in the process. It is required for converting one type of data from one analysis tool into another, but it is not used for any particular analysis. The arrows indicate the source and destination of the data, but the data is not necessarily conveyed in such a direct fashion. The output of DYMORE analysis is passed on to a MATLAB file or ModelCenter environment in order to be converted into the appropriate format as input data for the subsequent analysis tool.

3.2 Relational Design

Relational design is defined as method of linking part and product designs within a product structure with capabilities of parametric design and creation of parent/child relationships to control behavior. This definition is explained further in the following paragraphs.

Modern design practices of complex products such as rotorcraft involve hundreds of engineers, designers and experts of several design disciplines. This entire workforce

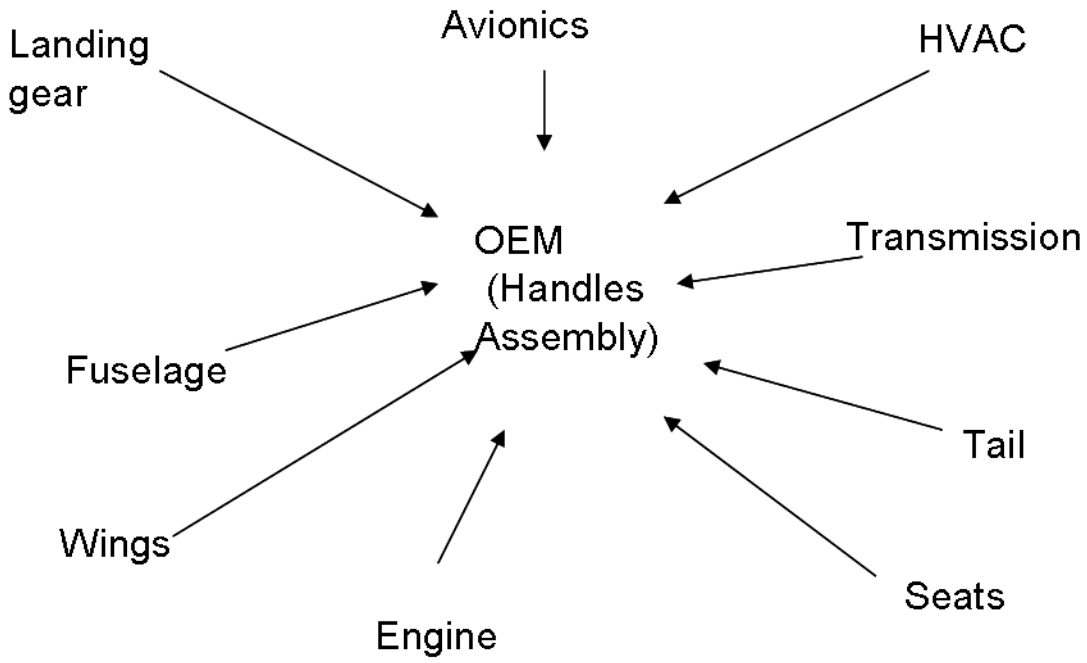


Figure 18: Distribution of work

is not necessarily employed by one company. In fact, the company which manufactures and markets the end product may make contracts with other companies to design and produce components of the product relatively independently. Figure 18 shows how the work distribution would look like for the case of aircraft. It is a simplified depiction of a particular design method; therefore not all subsystems of an aircraft are represented.

For the purpose of clarity, let the company which does assembly in Figure 18 be called Original Equipment Manufacturer (OEM). The other companies supplying subsystems will be supplier companies, or suppliers.

The OEM performs assembly; hence it has to ensure that all the parts and sub-assemblies fit together within specified tolerances. It is also the duty of suppliers to design and manufacture their products according to the instructions from the OEM, so that their product can be attached to the main assembly without problems. In order to meet the usually very tight aircraft tolerance requirements, the OEM must

inform its suppliers about the outer dimensions and/or attachment details of the particular products they are responsible of manufacturing. This task is best realized by providing them with CAD files containing exact geometrical limits, position coordinates, surface shapes, tolerancing and annotations and key features of the assembly areas. There may be a several CAD files to store all of this data. Some of them are defined below [17].

Major Dimensions File (MDF): This file supports structural arrangement and configuration of parts and subassemblies. It defines position and interfaces of detail parts. MDF is usually composed of points, lines and planes to be utilized as reference entities.

Surface Definitions File (SFD): It defines theoretical shape of the part or subassembly. Several MDS files constitute source of shape definition for all child products and downstream processes.

Product Relations Geometry (PRG): This file contains the geometry used to define and coordinate interfaces between two or more parts/subassemblies.

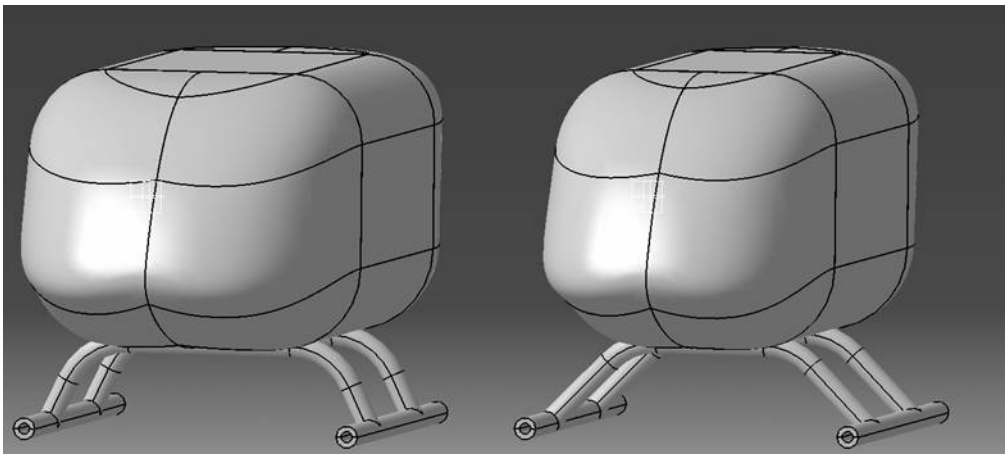
Installation Management File (IMF): It defines the list of installed part instances, together with tolerancing and annotation information for installing the parts.

The set of CAD files with key design information, including the ones defined above will be called datum files.

If, for example, a rotorcraft design is to be carried out using relational design concept, the MDF would include reference planes and geometry for structural members of fuselage, blades, hub, tail, control surfaces, and landing gear. It might also have airfoil definition for the blade and control surfaces. SDF file, on the other hand, would define surfaces such as blade, fuselage, tail, control surfaces and landing gear skins. The thickened skins and solid structures are generated in a separate part detail file derived from MDF and SDF files. Assembly of fuselage, tail, hub, etc. together would be achieved by using the information given in PRG and IMF.

Rotor design is inherently an iterative process. The design evolves during and after every analysis step of the iteration. This evolution naturally results in design changes. Every time a change occurs in the design, it requires same analyses and optimization studies, which may lead to more changes. This chain of actions creates a loop which ends when the design converges, i.e., no further changes are required at the end of an iteration. Relational design concept is very suitable for complicated systems such as rotorcraft, especially when it is vital for the OEM to share information with suppliers about the latest details regarding the situation of the product development process. If a design change occurs, the core company may instantly inform related suppliers about the modifications by sending them the new datum files. All the supplier company needs to do is to incorporate the new datum files into its product design files. Since the entire part or subassembly is derived from the datum files, as soon as those files are replaced, all the parts using the datum files as reference will be updated automatically. No human interaction is required, thus the updating process costs little time and money, enabling lower lifecycle costs of the entire product.

For instance, consider a rotorcraft company designing a helicopter with suppliers responsible for fuselage and landing gear. The fuselage and landing gear are modeled with reference to datum files. Toward the end of the design process, the OEM discovers that the helicopter needs to weigh less, and to accomplish that goal the fuselage has to be narrower. If those companies exercised traditional design methods, the fuselage manufacturer would have to redesign the subassembly. However, with all the CAD documents linked to datum files, the entire assembly may be updated almost instantly as soon as those files are fed into the database. Figure 19 illustrates the design change. Note that the landing gear has also been updated automatically.



(a) Before (b) After

Figure 19: Design change: Before (left) and after (right)

3.3 Optimization Procedure

The optimization of aircraft begins with general concepts of a numerical optimization problem that is defined by an objective function and constraints, all of which depend on design variables. The general form of an optimization problem is

$$\begin{array}{lll} \text{Minimize objective} & F(\vec{X}) & \\ \text{Subject to constraints} & g_j(\vec{X}) \leq 0 & j = 1, 2 \ldots J \\ & h_k(\vec{X}) = 0 & k = 1, 2 \ldots K \end{array}$$

where $\vec{X} = (x_1, x_2, \ldots x_p, \ldots x_n)$ is the vector of independent design variables, and $F(\vec{X})$ is the objective function. The functions $g_j(\vec{X})$ and $h_k(\vec{X})$ are inequality and equality constraint functions, respectively. This simple numerical formulation was recognized by Schmit [89] for its applicability to the engineering design problem in 1960. Based on this discovery, Stepniewski et al. suggested the application to helicopters in 1970 [90]. However, the formulation was not widely adopted by the rotorcraft industry until the early 1980's due to the complexity of the rotorcraft optimization problem itself, even when a simple formulation was considered. In order

to achieve a truly optimum design, all the design disciplines should be considered at the same time. Additionally, other aspects, such as life-cycle cost, manufacturing, mission profiles and maintainability, need to be accounted for. Several issues to be optimized usually introduce conflicting objectives. Furthermore, the design space is rarely unimodal, the number of design variables is not manageable with manual approaches, and analysis tools often give results that differ from test results.

Probably the most effective way of dealing with aircraft optimization is using Multidisciplinary Design Optimization (MDO) principles. There are a number of formal definitions of MDO. It is defined by Simpson [91] as a methodology for design of systems with strong interactions between its disciplines, such that the variables in several disciplines need to be manipulated simultaneously. MDO involves the coordination of multiple disciplinary analyses during the design and optimization of complex systems.

Application of MDO principles into helicopter design has been successfully performed by Khalid [92] in 2006. He developed a framework for preliminary rotorcraft design using IPPD methodology. He included all the technical aspects of design such as vehicle engineering, dynamic analysis, stability and control, aerodynamic performance, propulsion, transmission design, weight and balance, noise analysis and economic analysis for minimizing total cost of a training helicopter. A schematic representation of his framework is depicted in Figure 20. All the grey boxes under the optimizer are called contributing analyses (CA's). They represent sets of tools to perform analysis or optimization locally.The result of those analysis are then sent to system optimizer. Although Khalid incorporated several design disciplines into the global optimization problem, he did not perform high fidelity analysis in most of the disciplines. In order to have a complete design system, all the disciplines shown in Figure 20 need to have their own high fidelity analysis and/or optimization process. In order to partially meet this requirement, this study focuses on a portion of

dynamic and structures discipline, namely natural frequency analysis and dynamic stability behavior of the rotor system.

Although the interest of this research is in a piece of the global rotorcraft design, achieving the optimum solution for a given objective is still not an easy task. Even for the small subsection of a CA, there are numerous parameters that define structural and dynamic properties of a rotor. The design space for an objective function that is defined by those parameters therefore has multiple dimensions, implying the design possibilities being represented in hyperspace. It may be difficult or impossible to visually inspect such a design space and point out the optimum value. Moreover, one of the goals of this research is automating the design and optimization. Thus a mathematics and/or statistics-based optimization method is necessary.

Among the mathematics-based optimization methods, gradient-based methods are effective, and most commonly used. Using an appropriate computer system, it is possible to reach to the global or local minimum of a function with an accuracy of $\pm 10^{-6}$. On the other hand, the results depend highly on the initial point of search directions. If the function is not unimodal it would be quite a challenging task to achieve global optimum using only gradient-based methods. Moreover, calculating gradient of a function needs several function calls, which might be expensive for if along analysis time is required for each function evaluation.

The rotor system under consideration has many design variables, and it is possible that the design space for almost any given objective function $F(\vec{X})$ has lots of local minima, like a distorted egg crate. For ease of visualization purposes, a possible design space for two design variables and one objective function is depicted in Figure 21. The horizontal axes represent the two design variables, and the vertical axis represents the objective function. The values on the axes are meaningless, since this plot is presented for visualization only. If an existing design is to be optimized, the configuration (i.e. current values of all design variables) of the current design is also

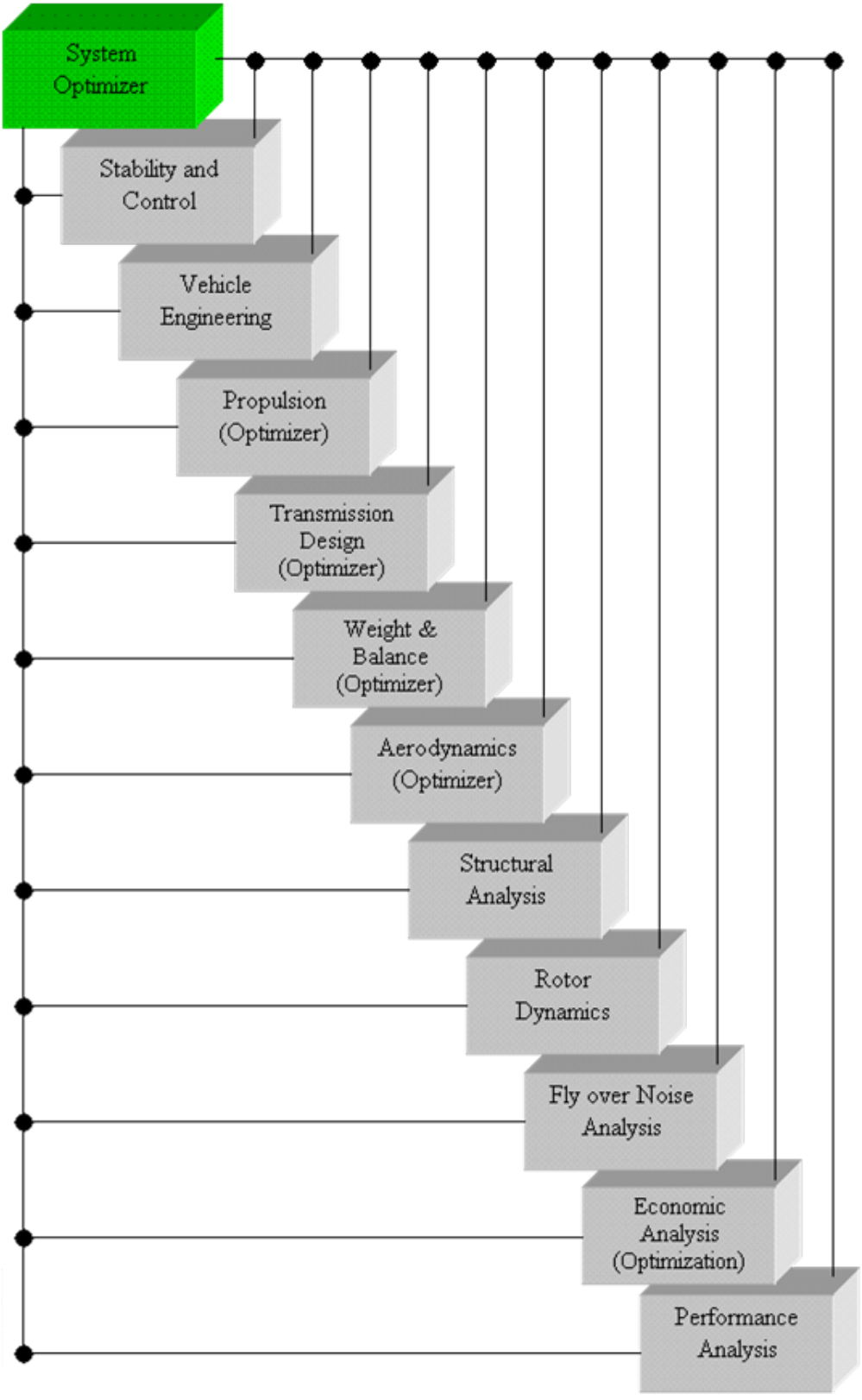


Figure 20: Design disciplines Khalid included in his framework [92]

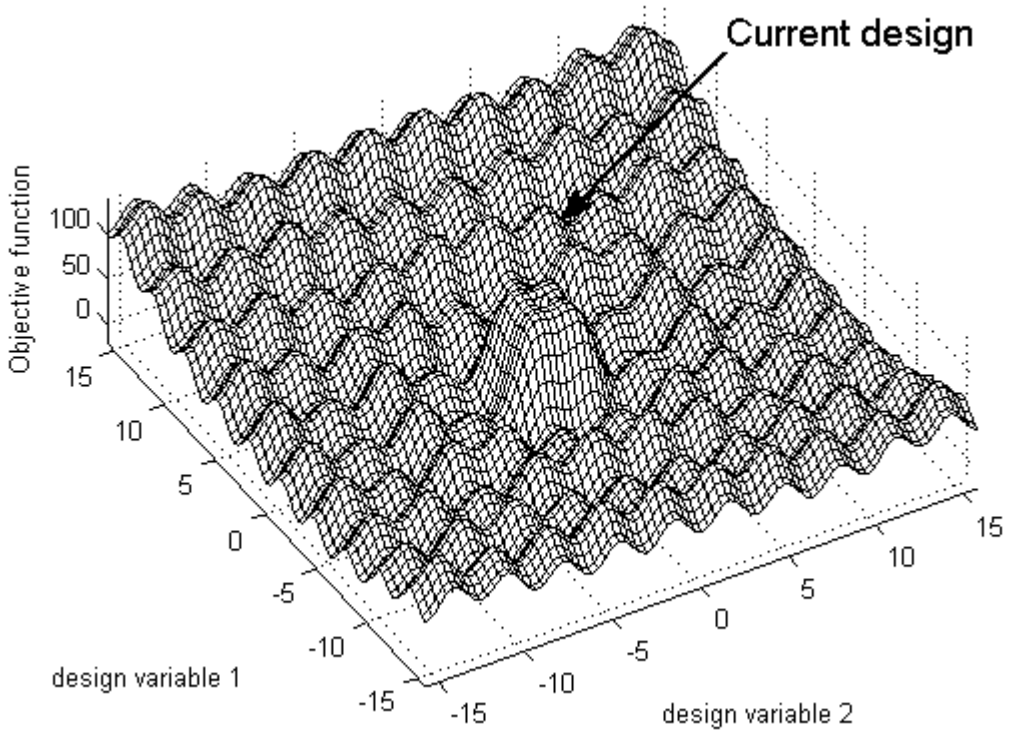


Figure 21: A representative design space for two design variables

in the same space.

For any given objective function $F(\vec{X})$, it is assumed that there are no constraints ($g_j(\vec{X})$ or $h_j(\vec{X})$) on the design variables at this time. Effect of constraints are discussed later in this section. For the case of having no constraints on the problem, there are a number of techniques suitable for optimization. For continuous but complicated design spaces such as the one in Figure 21, one can use grid search, random search, random walk, genetic algorithms (GA) and simulated annealing (SA) among a wide selection of optimization methods. Those methods may produce a result close to the true optimum, but they highly depend on initial point or points chosen by the optimizer, and also randomness is involved during the optimization process. A good result might not be achieved if the optimizer makes unlucky decisions. In order to reach the optimum with more confidence, a multi-step procedure is considered. Such a procedure is described below.

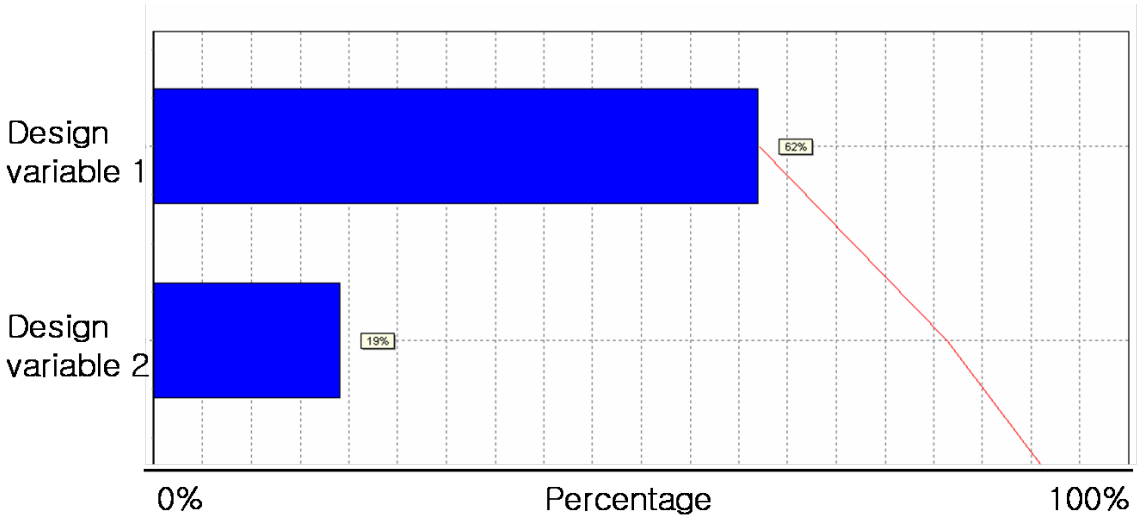


Figure 22: Identification of the most important design variables

Step 1. Identification of the most important design variables. According to the Pareto Principle, among the many design variables that define the objective function, only a few of them influence the change in objective function the most. Identifying the most important and the least important design variables beforehand may help simplify the optimization computations. ModelCenter is utilized to perform this task. The result may look like the bars in Figure 22. There may be more than two design variables in the actual problem; the figure is for demonstration purposes only.

Step 2. Performing a preliminary optimization with the most important design variables. It is assumed that the most important design variables define the neighborhood of optimum objective function. In other words, if the $F(\vec{X})$ objective reaches a minimum or a maximum (whichever is desired) for a specific value of the most important variables, say $\vec{x_p}$, the true optimum is in the vicinity of that particular value of $\vec{x_p}$. The remaining design variables may be tweaked later to approach to the global optimum further. For design space with discontinuous variables, the optimization study may be conducted by means of methods like genetic algorithms, or sampling the design space at varying values of $\vec{x_p}$ between its upper and lower limits, while keeping the other variables constant. Figure 23 shows representative results for random

sampling. It is preferrable to genetic algorithms if evaluation the objective function is costly. In the case of the figure, of the two design variables, one of them is kept constant while the other one is varied between its limits. The white rectangle implies the constant behavior of the less important variable at a value of six. Using the design variable value combinations and their corresponding objective function results, a response surface is generated. This new surface, say $F_{rs}(\vec{X})$, replaces the original design space in the subsequent steps in order to find its optimum with less computational effort. This response surface may be of order two or higher, depending on method of subsequent optimization method. For approximating higher order surfaces, kriging method may be used.

Step 3. Finding the global optimum for continuous objective functions. The optimum point found in the previous step is the optimum of the approximate function, $F_{rs}(\vec{X})$. It is not necessarily the optimum of the actual objective function $F(\vec{X})$. However, the global optimum is possibly in the neighborhood of approximate optimum. A line search starting at the approximate optimum probably reaches to global optimum, as depicted in Figure 24.

Of the three steps discussed above, one or two of them may be omitted dependong on the nature of the optimization problem. If there are not many design variables, for instance, the step of identifying the most important ones may be redundant, hence can be skipped. In some cases, the optimum configuration can be guessed to be near a particular area in the design space, therefore, it may not be necessary to perform step 2; that is, the next step is performed with a configuration expected to be already close to the global optimum. Finally, if the design space is known to be discontinuous, it may not be possible to perform the last step. In that case, performing step 2 with more samples may produce results closer to the global optimum.

In case of inequality constraints on the design variables, the design space may be confined to the limitations of the constraints, hence at least one of the limits of

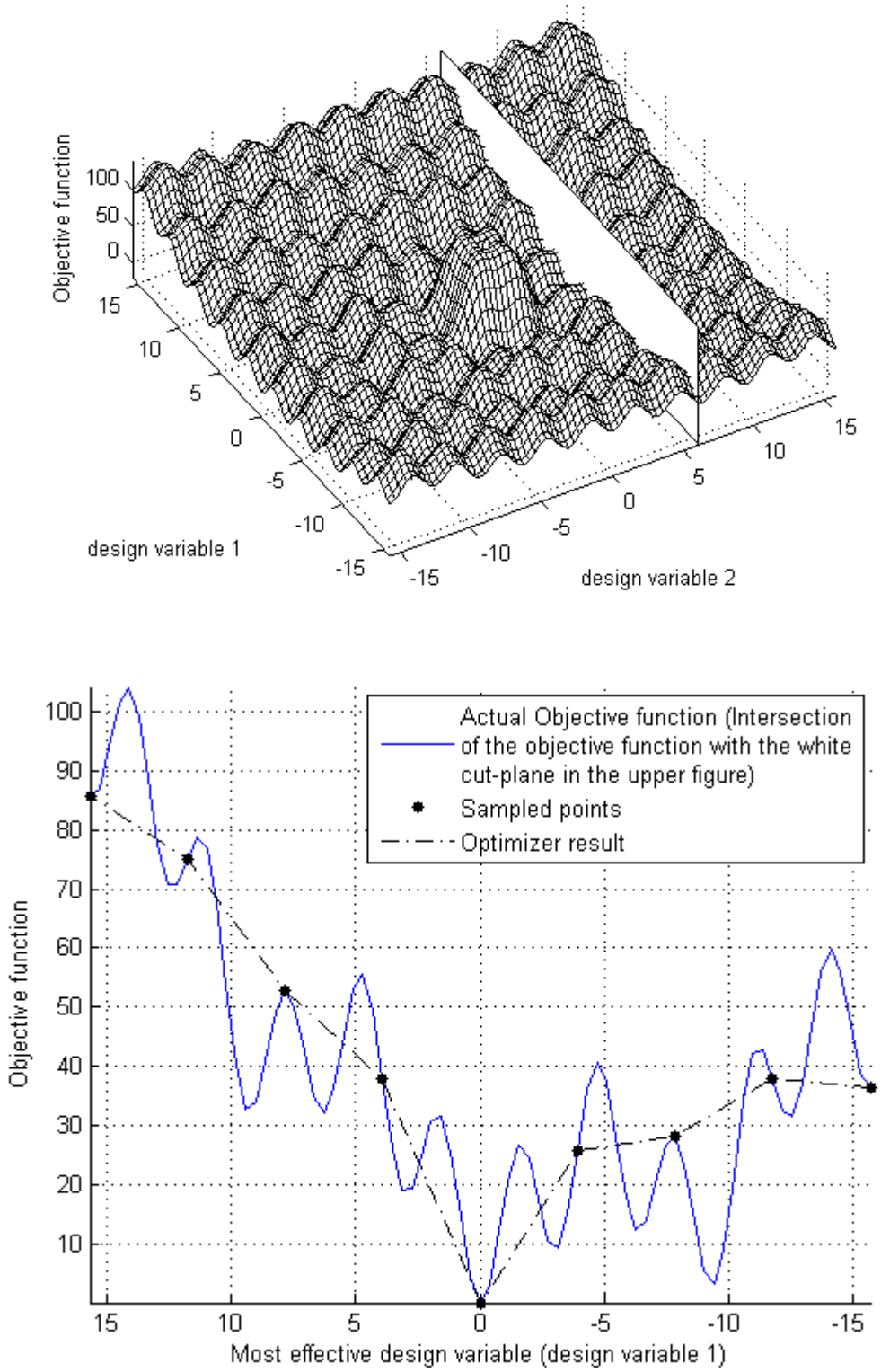


Figure 23: Performing a parametric study with the most important design variable

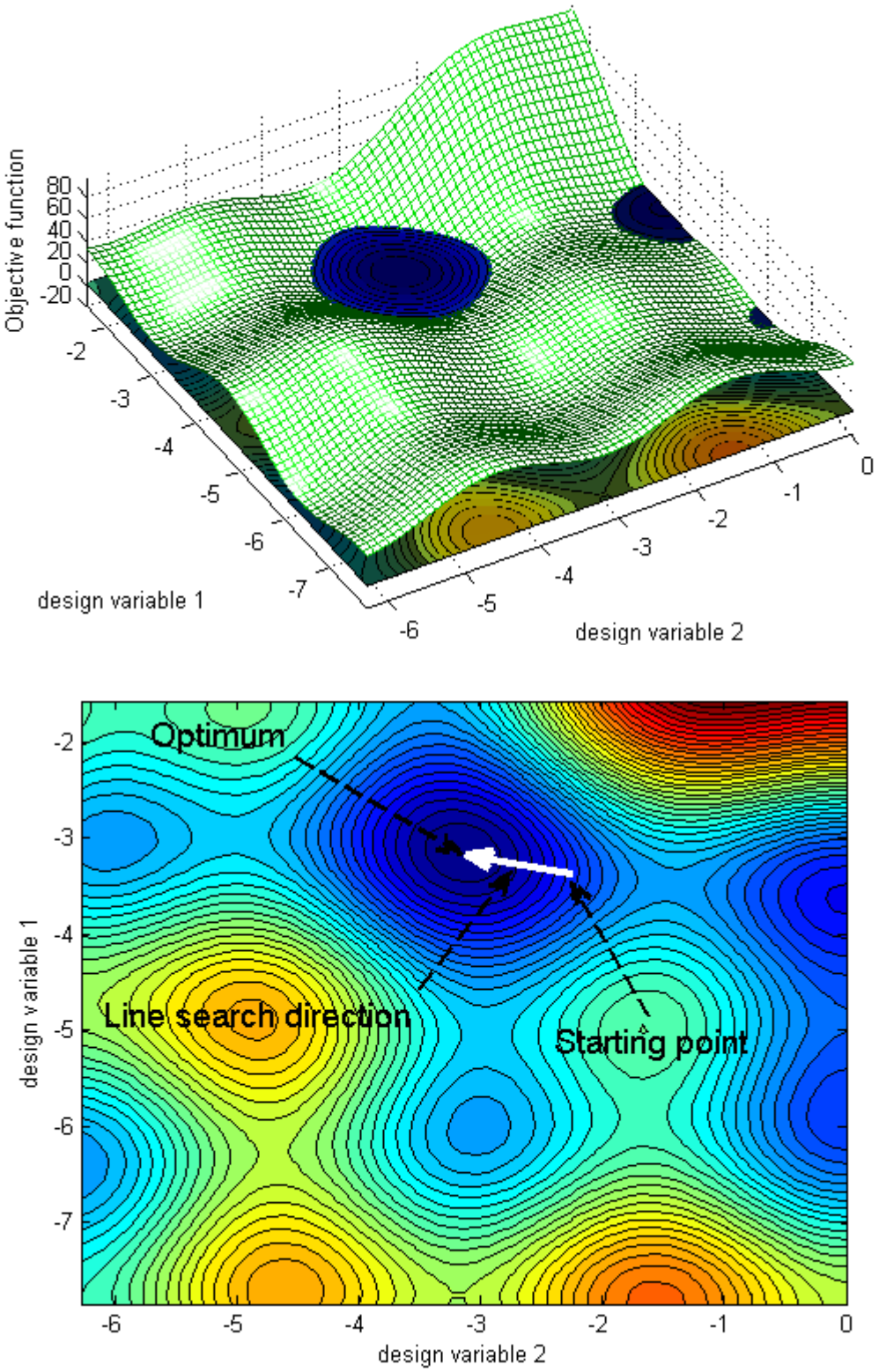


Figure 24: Finding the global optimum using a line search method

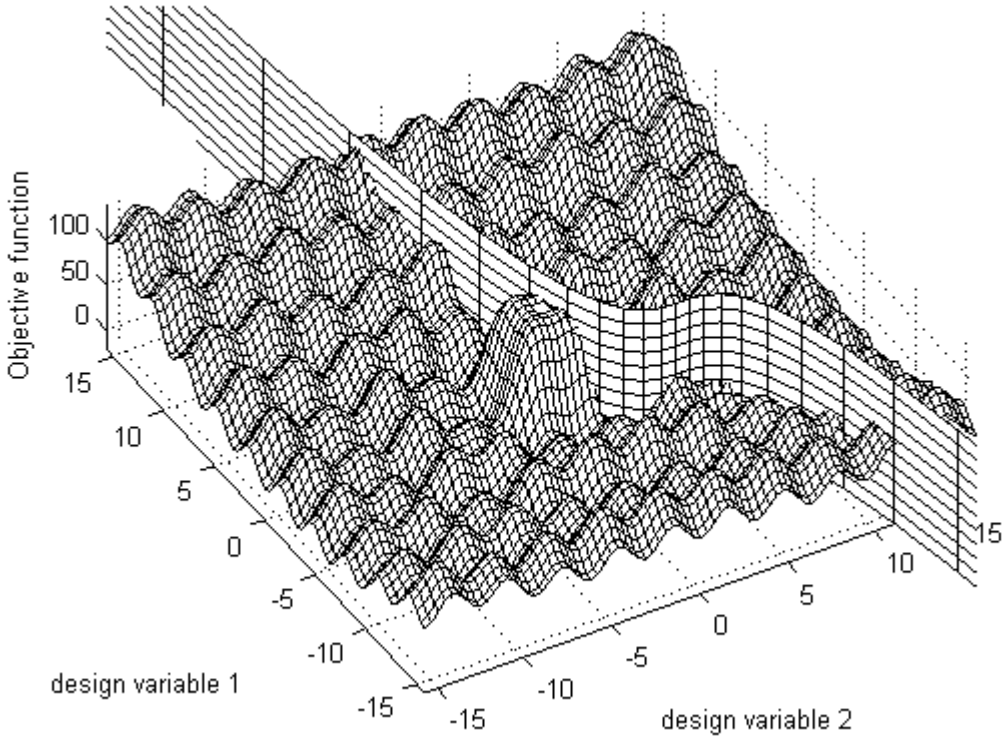


Figure 25: Design space with a constraint

the design variables are dictated by the inequality constraints, as depicted in Figure 25. Following this adjustment, rest of the steps described above may be followed to reach the optimal design. Equality constrains may be used to eliminate some design variables from optimization problem altogether. This reduces the number of design variables to vary, and in turn, dimensions of the design space.

The optimization method presented here is applied to rotor system optimization portion of the CA "Structural Analysis" in Figure 20. The entire optimization of the rotorcraft is carried out by collaboration of all the other disciplines working in tandem. The process in Figure 20 is hence called "collaborative optimization" (CO).

CHAPTER IV

APPLICATION OF PROPOSED METHODOLOGY

For application of the optimization method described in the previous chapter, Thomas Hanson's bearingless Auto-Trim rotor system is considered. The schematics of the system are provided in Figures 26 and 27 [69]. The blade and flexure are modeled in CATIA. The CATIA model is given in Figure 28.

The rotor system is mainly comprised of rotor blades, flexure beams, hub, torque tube assembly, and pitch-link assembly. The flexure beams replace all hinges and bearings of the existing articulated rotor. These flexures allow flapping, feathering, and lagging motion. The dynamic characteristics of the presented hingeless rotor are significantly affected by the sizing and geometry of flexure beams. Hence, the appropriate sizing of flexures is the key element for a successful rotor system. The size of flexure is computed by using an Excel spreadsheet originally developed by Luke Priesner at Georgia Tech, based on Thomas Hanson's Designer Friendly Handbook [69]. More information with general dimensions and angles are provided in Table 1.

Auto-Trim rotor has no hinges, but its flexure beams are soft enough to allow the blades to move flapwise and chordwise directions, as well as feathering motion. The blades behave as if there is an effective hinge offset at about 10% of the rotor radius away from the center of the hub. This amount of hinge offset is more than the usual percentage of 3 or 4% in today's rotorcraft, except for the Sikorsky X2, former MBB hingeless rotors, BO105 and BK117, which have effective hinge offsets of 15-20%. This difference enables more damping for fuselage oscillations and more control power on the rotor by the pilot. On the other hand, a rotorcraft with a 10% hinge offset would be more sensitive to wind gusts and other disturbances to the rotor,

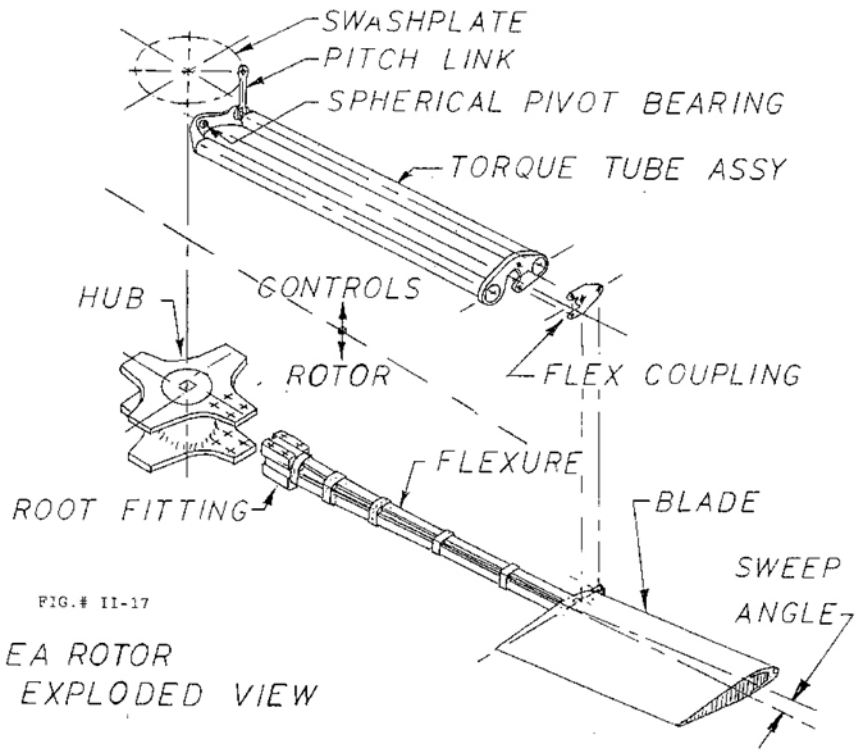


Figure 26: Schematics of Hanson's blade and flexure system

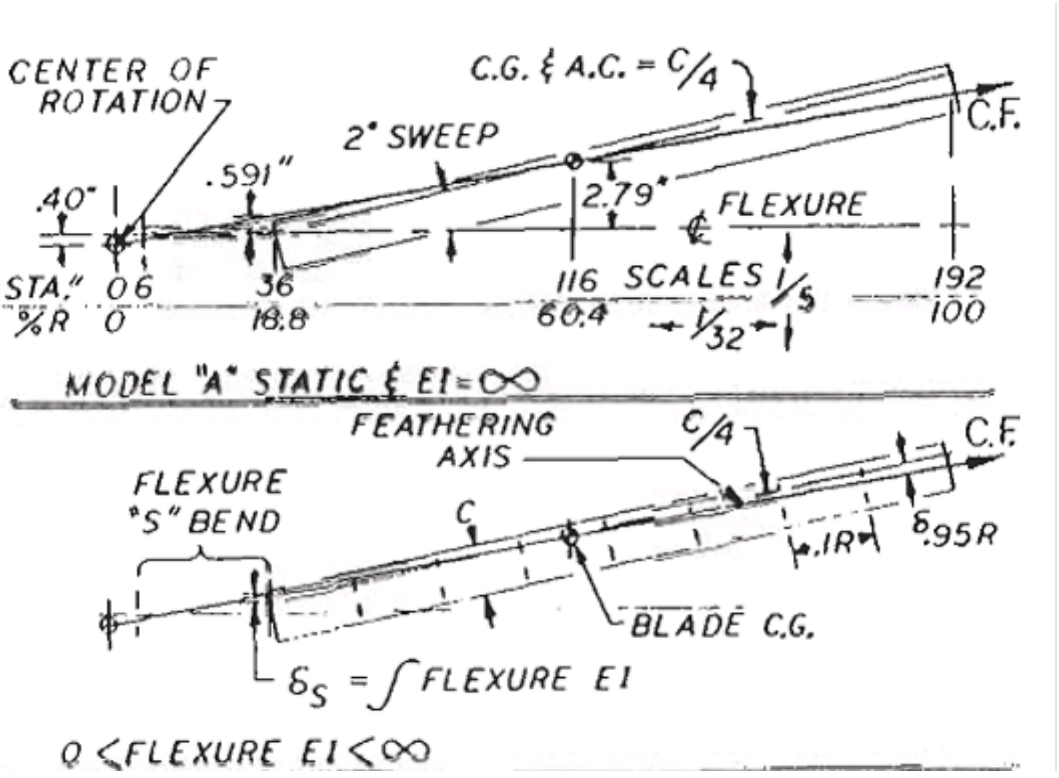


Figure 27: Hanson's forward swept rotor blade

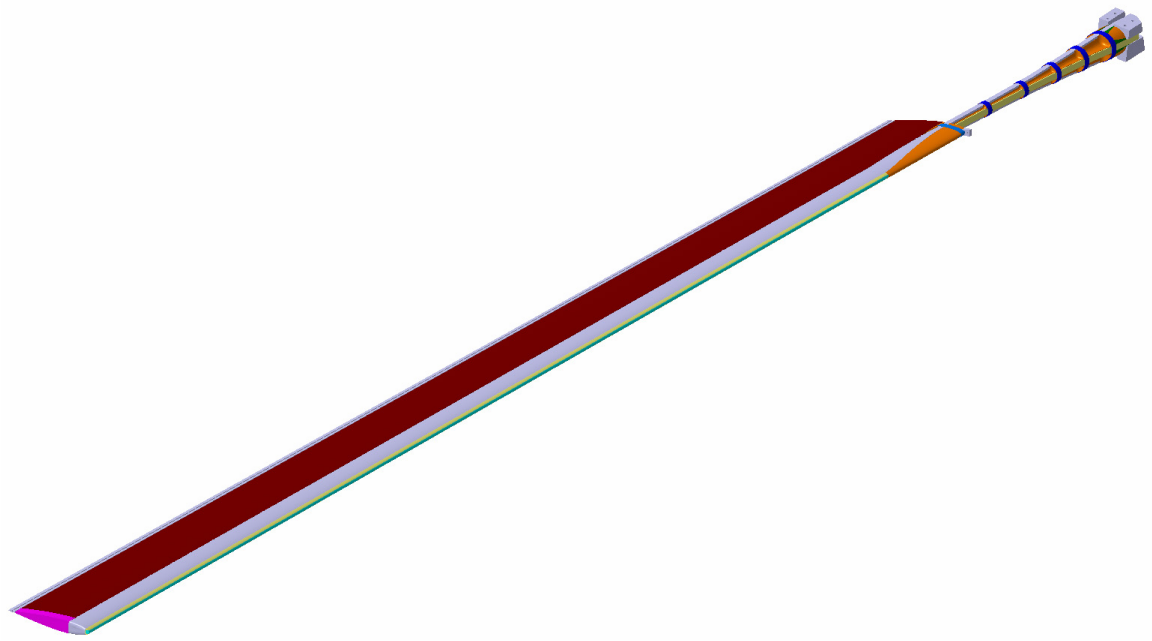


Figure 28: Hanson's rotor blade and flexure model in CATIA

Table 1: Auto-Trim rotor system configuration

Parameter name	value	unit
Rotor radius (R)	18.0	ft
Blade chord length (c)	11.0	in
Blade operating angular velocity (Ω)	33.33	rad/s
Blade precone angle (β_0)	3.5	deg
Blade forward sweep angle (ζ)	2	deg
Blade linear twist angle (θ_{tw})	12	deg
Blade tip mass (m_{tip})	0.100144	slug
Blade taper	(none)	
Airfoil type	NACA 0015	
Blade tip dihedral	(none)	
Blade tip sweep	(none)	

since more rotor moments are transferred to the fuselage, and the pilot would have to correct for these disturbances. In order to couple flapping with feathering and reduce control force inputs, the rotor blades are swept forward by a few degrees. As a result of forward blade sweep, when an external effect changes the lift force on the blades, the blades pitch in order to modify the angle of attack to create a force opposing the change. This relationship between flapping and feathering motions alleviate effect of gusts almost immediately. In order to make this system work properly, the first flapping and first feathering natural frequencies of the rotor must coalesce and come close to each other.

As mentioned in Chapter 3, the Auto-Trim rotor system does not need hydraulic or any other boosting mechanism to augment pilot control inputs, because the control inputs are at resonance with the flapping and rotational speed, thus minimizing the input force. At operating angular velocity, the first flapping and feathering natural frequencies are close to 1P (per revolution), which is also swash plate control input frequency. Exciting the rotor at its resonance frequency attenuates the force to be applied to the rotor, thus eliminating the need for power boost. Having a direct mechanical connection between the control stick and rotor also enables feedback to the pilot about the forces on the rotor. Theoretical background and other analyses on Hanson's rotor system are presented in the appendix.

4.1 Optimization Setup

One of the major advantages of the Auto-Trim system over other rotor designs is its stability characteristics under gust conditions. As mentioned earlier, the rotor was designed in 1960s. At the time, high speed computers were scarce, and design optimization in rotor blades was almost nonexistent. This fact gives way to the possibility that although the current design is effective, there may be room for further development. With this motivation, an optimization problem is set with few design

variables and an objective of minimizing time for flapping oscillations to fade.

The focus of this analysis is to minimize blade vibrations in order to improve blade life, and reduce hub vibrations which would be fed into the fuselage, rather than improving comfort of passengers in cabin. For this reason, research interest is more on minimizing the oscillations with the greatest amplitude at the blade tip, rather than minimizing loads and moments on rotor hub, which is usually the goal in rotorcraft vibration studies. Nevertheless, cabin vibrations might be alleviated as well as a result if this study. The reason for this expectation is that, the aerodynamic loading depends strongly on the blade shape and angle of attack. If the blade position and orientation does not change dramatically with respect to the hub under gust conditions, no unnecessary or unwanted forces will be generated.

The rotor operation is considered to be in hover at all times. According to ref. [30], effect of forward flight is usually aeroelastically stabilizing for soft-in-plane blade configurations.

The optimization problem at hand has constraints on design parameters. The natural frequencies of each oscillation must be either close to or away from particular forcing frequencies. In order to satisfy the ease-of-control requirement of the Auto-Trim rotor, first flapping frequency must be close to 1P. For achieving the stable response, first feathering frequency must be close to first flapping frequency at operating angular speed. All other angular frequencies must be away from integer multiples of 1P. Also, for a rotor with N blades, the N/rev forces and moments transmitted by the rotor to the fuselage constitute the primary source of vibration in the cabin. In addition, (N-1)/rev and (N+1)/rev motions are important for hub pitch and roll moments and for in-plane moments. The rotor under consideration has four blades, hence the important frequencies to avoid are 3P, 4P, and 5P.

The oscillations under consideration in this case are flapping, lead-lag and twisting at the blade tip. Figure 29 shows the time history the flap-wise displacement at

the blade tip for the initial design in hover condition. In this analysis, a twisting moment is applied at the tip at time t=0 seconds and then the blade is left to oscillate freely while rotating at nominal angular velocity. The flap-wise response due to the disturbance is stabilized.

In another analysis, the blade is accelerated to the nominal angular velocity from rest, causing a wake formation. Then, a constant wind gust is applied from one side in hover for one second, as a step function. Finally, the wind is removed, and the blade is allowed to oscillate freely, at the same angular speed. The resulting behavior is found to be stable, as shown in Figure 30. Other oscillations, namely lead-lag and pitching motions, have been found to be stable as well, as shown in Figures 31 and 32. See appendix for details on VABS and DYMORE analysis. This result and frequency analysis (i.e. fan plots) indicate that the current blade design is dynamically acceptable, therefore it is feasible.

Since the current design is feasible, its configuration presented in Table 1 may be used to assign limits for design variables, such that the initial configuration stays within the design space. Table 2 identifies design variables and their upper and lower limits. Another reason for setting limits close to the current design is that this configuration was obtained after consideration of several disciplines, not just structures. An optimum design with parameter values close to the existing configuration might still be acceptable for other discipline analyses. For instance, the lower limit for the rotor radius is set to 14 ft. The limiting factor for a low radius is the disk loading of the rotor. A low disk loading is desirable for helicopters. The gross weight of the helicopter for which this rotor is designed is 3395 lbs. A maximum disk loading of 6 lbs/ft^2 is assumed to be fit for this rotor, hence the corresponding minimum radius is 14 ft. Other than disk loading, there are many factors affecting the rotor diameter, such as hover performance, tip speed, noise and autorotation considerations. Since the analysis in this study is of demonstration purposes only, none of the

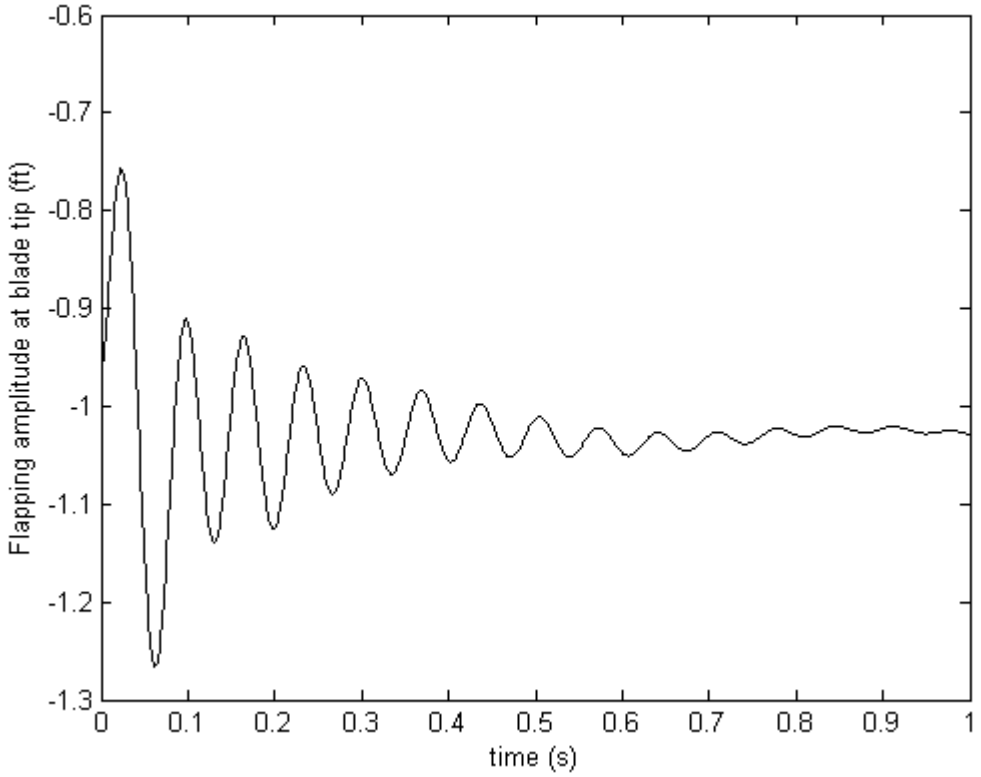


Figure 29: Flapping distance at blade tip after an initial twisting disturbance

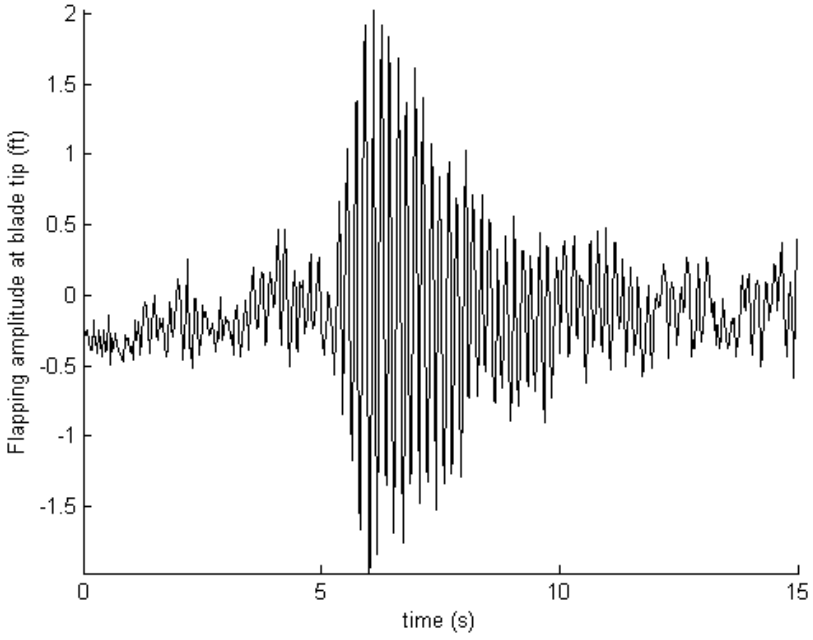


Figure 30: Flapping distance at blade tip after a wind gust disturbance at t=5s.

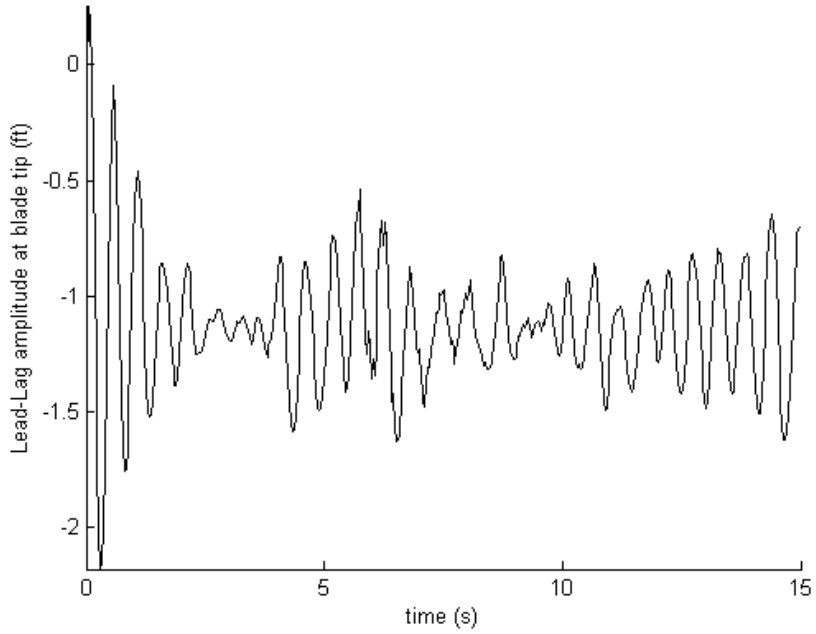


Figure 31: Lead-lag motion at blade tip after a wind gust disturbance at t=5s

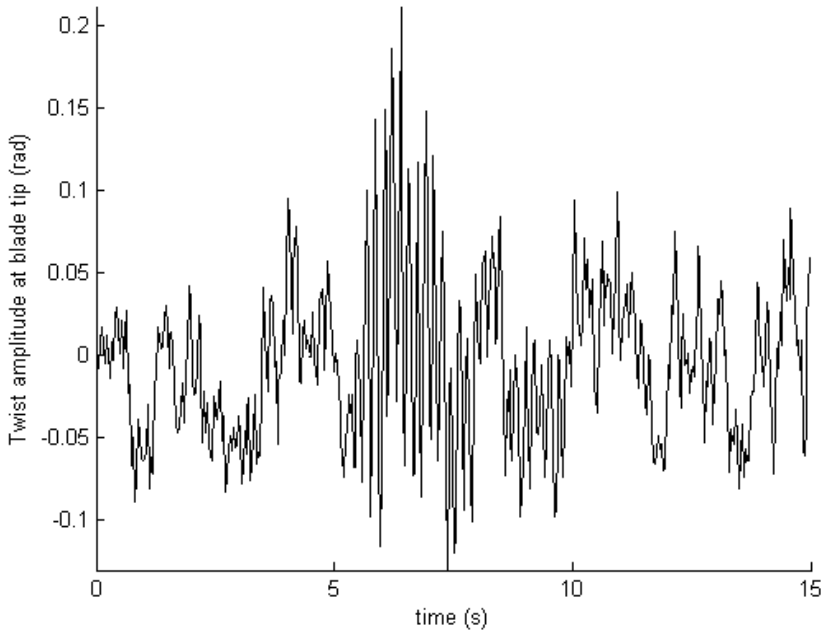


Figure 32: Pitching motion at blade tip after a wind gust disturbance at t=5s

Table 2: Design variables and limits

Design variable (parameter)	Lower limit	Upper limit	unit
Rotor radius (R)	14	25	ft
Rotor operating angular velocity (Ω)	28	38	rad/s
Blade chord length (c)	9	13	in
Blade forward sweep angle (ζ)	0.001	4.0	deg
Blade linear twist angle (θ_{tw})	8.0	16.0	deg
Blade tip mass (m_{tip})	0.001	0.20	slug
Flexure arm width	0.3	1.2	in
Flexure arm thickness	0.1	0.4	in

upper and lower limits of any design variable are strictly defined. In order to make sure the optimization is concluded at the global minimum of the objective function, and not bounded by a constraint, the limits of all design variables are established to cover the widest design space possible. Likewise, since this is a demonstration case, in order to achieve a configuration that will satisfy the objective more effectively and produce dramatically different results from the original configuration, most major design variables were included as design parameters, although they would probably be strictly constrained if not fixed by design disciplines that are not incorporated in the optimization loop. The limits for the rotor radius, for instance, are unrealistically relaxed for a single main rotor helicopter such as the one considered in this study.

The objective function is chosen as to minimize flapping, lead-lag and pitching oscillations, and the constraints are defined as specific frequencies to avoid or to achieve. For instance, the rotor design under consideration here must have the first flapping frequency f_{flap1} close to $1P$ and first lead-lag frequency $f_{lead-lag1}$ near $0.5P$. The second flapping frequency, f_{flap2}, however, should be away from all integer multiples of the rotational frequency, P. Together with the selected design variables, the optimization problem may be stated as follows:

Minimize objective $\quad F(\vec{X}) = S_{flap} + S_{lead-lag} + S_{pitch}$

Subject to constraints:

$$g_1 = f_{flap1} - 1.2 \cdot P \leq 0$$

$$g_2 = P - f_{flap1} \leq 0$$

$$g_3 = 0.4 \cdot P - f_{lead-lag1} \leq 0$$

$$g_4 = f_{lead-lag1} - 0.6 \cdot P \leq 0$$

$$g_i = f_i - k \cdot 0.95 \cdot P \leq 0$$

$$k \cdot 1.05 \cdot P - f_i \leq 0$$

$$i = 5, 6, 7, \ldots \qquad k = 1, 2, 3, 4, 5$$

where S_{flap}, $S_{lead-lag}$ and S_{pitch} are measures for blade flapping, lead-lag and pitching response stabilities, respectively. The stability S is a nondimensional quantity, defined as the product of the time when amplitude of the response reduces to 50% of maximum amplitude ($t_{50\%}$) in seconds, with the sum of 20 times the amplitude (d) between $t = 12$s and $t = 15$s and the maximum amplitude of the oscillation (A_{max}), in feet. It is nondimensionalized by multiplying a constant, k, defined as $1/(ft \cdot s)$. Clearly, the time and amplitude values are implicit functions of the design variables identified before. Figure 33 explains the details. In equation form,

$$S_{flap} = t_{50\%} \cdot (20 \cdot d + A_{flap}) \cdot k$$

$S_{lead-lag}$ and S_{pitch} are defined in the same manner. In the optimization problem definition above, f_i is the i^{th} nondimensional natural frequency in multiples of revolution. It could be flapping, feathering or lead-lag, except for the first flapping or first feathering frequency. P is the operational frequency of the rotor.

For simplification of the problem, the constraints can be incorporated into the objective function as follows:

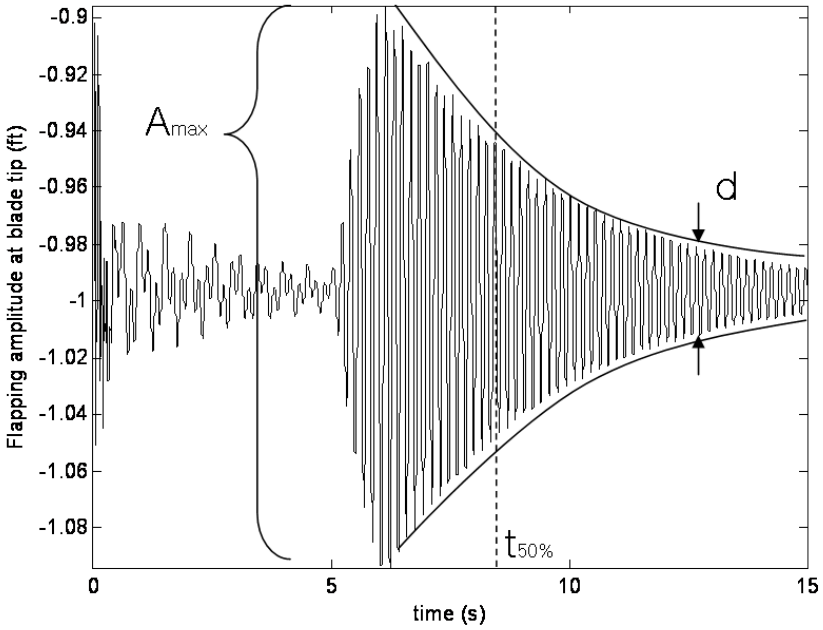


Figure 33: Definition of terms for calculating stability measure S_{flap}

$$\begin{aligned} F(\vec{X}) =& S_{flap} + S_{lead-lag} + S_{pitch} + (1 - flap_1) + (flap_1 - 1.2) \\ &+ (0.4 - leadlag_1) + (leadlag_1 - 0.6) + \frac{1}{|T_{f2} - flap_2|} \\ &+ \frac{1}{|T_{f3} - flap_3|} + \frac{1}{|T_{f4} - flap_4|} \end{aligned}$$

where T is an integer closest to the particular natural frequency. All frequencies are in [per rev].

Optimization is performed using a detailed version of the the existing DYMORE model given in Figure 55. The detailed model is shown in Figure 34. In a traditional optimization procedure, it would be acceptable to conduct a line search type of optimization, starting with the current design, since it was found to be already feasible. However, the possibility of having a complicated, multimodal design space with several local minima and maxima, instead of a unimodal one - with only one global optima, raises the need for a different approach. In past research, blades to be optimized usually had simple crossections throughout their spans, such as the one

shown in Figure 35. The Auto-Trim rotor blade, on the other hand, has a complicated geometry with different types of materials, including composites with different lay-up angles.

The optimization problem is setup in ModelCenter a shown in Figure 36. The DYMORE model is defined within specific text files with extension *.dym* for DYMORE to process. Those text files are generated in ModelCenter through a wrapper file. Whenever a change occurs in a design variable, new information is entered into this wrapper file. ModelCenter then generates input files for DYMORE analysis. At the end of dynamic and static analysis, DYMORE generates result files. Those results are read by MATLAB for data extraction and analysis within the code *stability* in the figure. New design variables and objective function values are imported from MATLAB to ModelCenter optimizer for the next iteration. All the wrapper files, DYMORE input files, and ANSYS macros for this research are given in Appendix E.

4.2 Optimization Procedure

The optimization steps discussed in Chapter 3 are implemented to the Auto-Trim rotor system optimization problem.

4.2.1 Performing a parametric study and identification of the most important design variables

Since there are few design variables, identification of the most important design variables is not absolutely necessary for this particular case. However, for demonstration purposes, the eight design parameters defined in the previous section are varied within their limits using Latin Hypercube Sampling. After sixteen system runs with different combinations of the design variables, the variables blade root angle of attack and blade forward sweep angle ζ are found to be the most influential parameters among the eight parameters as shown in Figure 37.

At the end of the parametric study the design with the lowest objective function

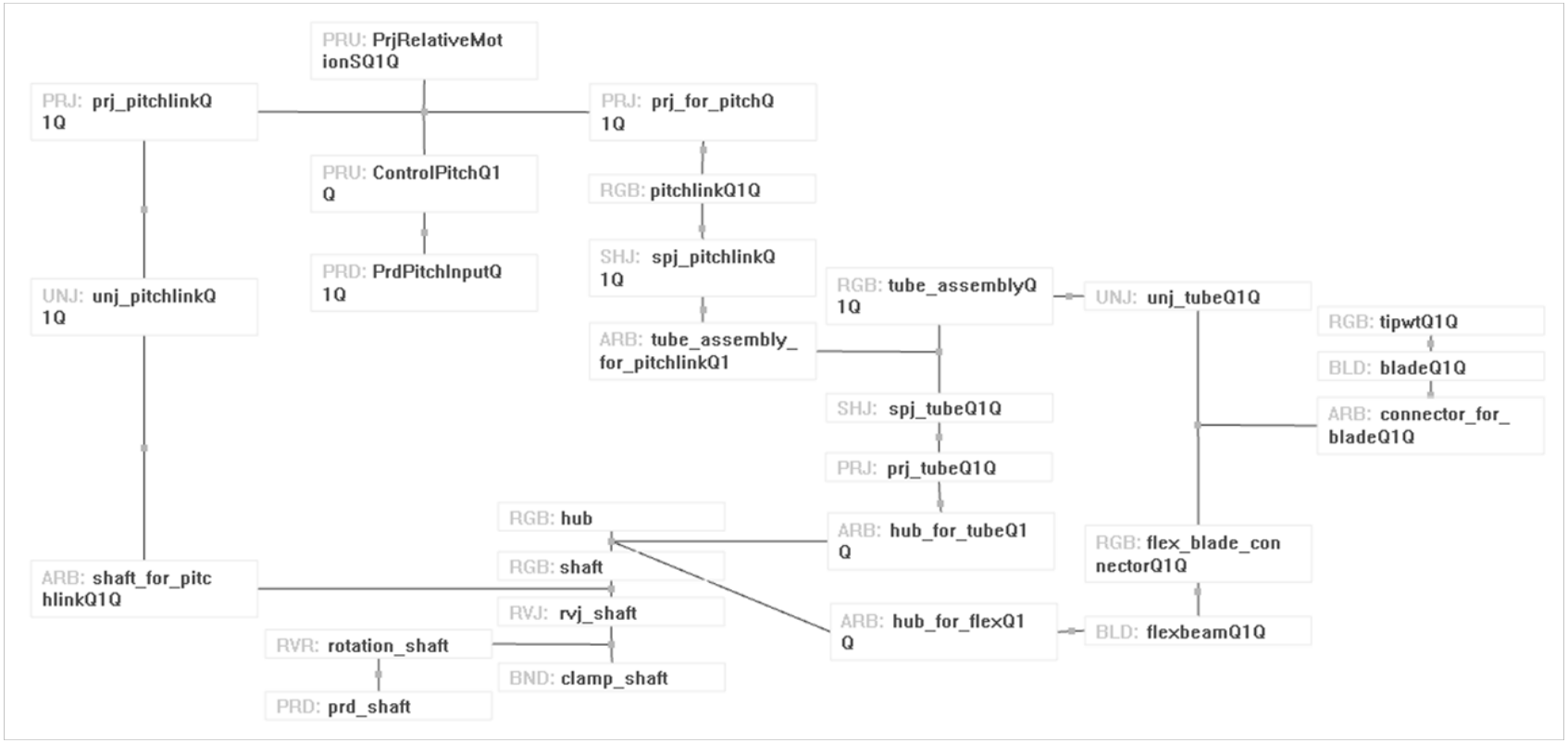


Figure 34: Detailed DYMORE model of the rotor system

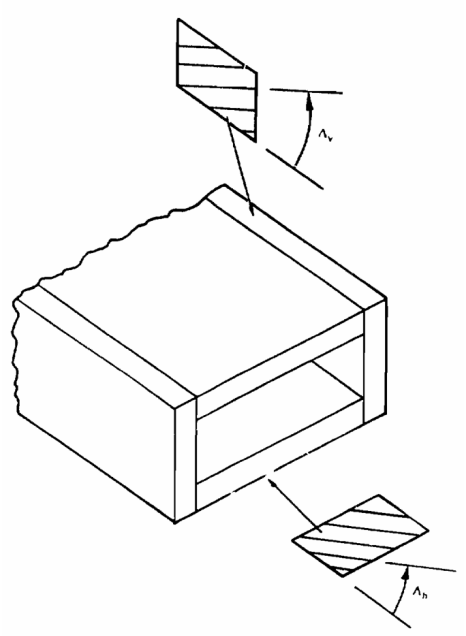


Figure 35: A simple blade crossection from Ref. [39]

value was found to have the configuration given in Table 3.

4.2.2 Finding the global optimum

Using a line search algorithm called "Conjugate Gradient Method" in ModelCenter, optimum is reached in three steps. However, the time-intensive codes ANSYS and DYMORE had to be run many more times, including the runs necessary for calculating gradients. Only the two parameters, namely, forward sweep angle and chord length, identified previously as important are varied. The initial values are

Table 3: Configuration at the end of the parametric study

Design variable (parameter)	value	unit
Rotor radius (R)	14.1673	ft
Rotor operating angular velocity (Ω)	32.81	rad/s
Blade chord length (c)	11.9899	in
Blade forward sweep angle (ζ)	2.6	deg
Blade linear twist angle (θ_{tw})	8.99	deg
Blade tip mass (m_{tip})	0.1109	slug
Flexure arm width	0.452	in
Flexure arm thickness	0.209	in

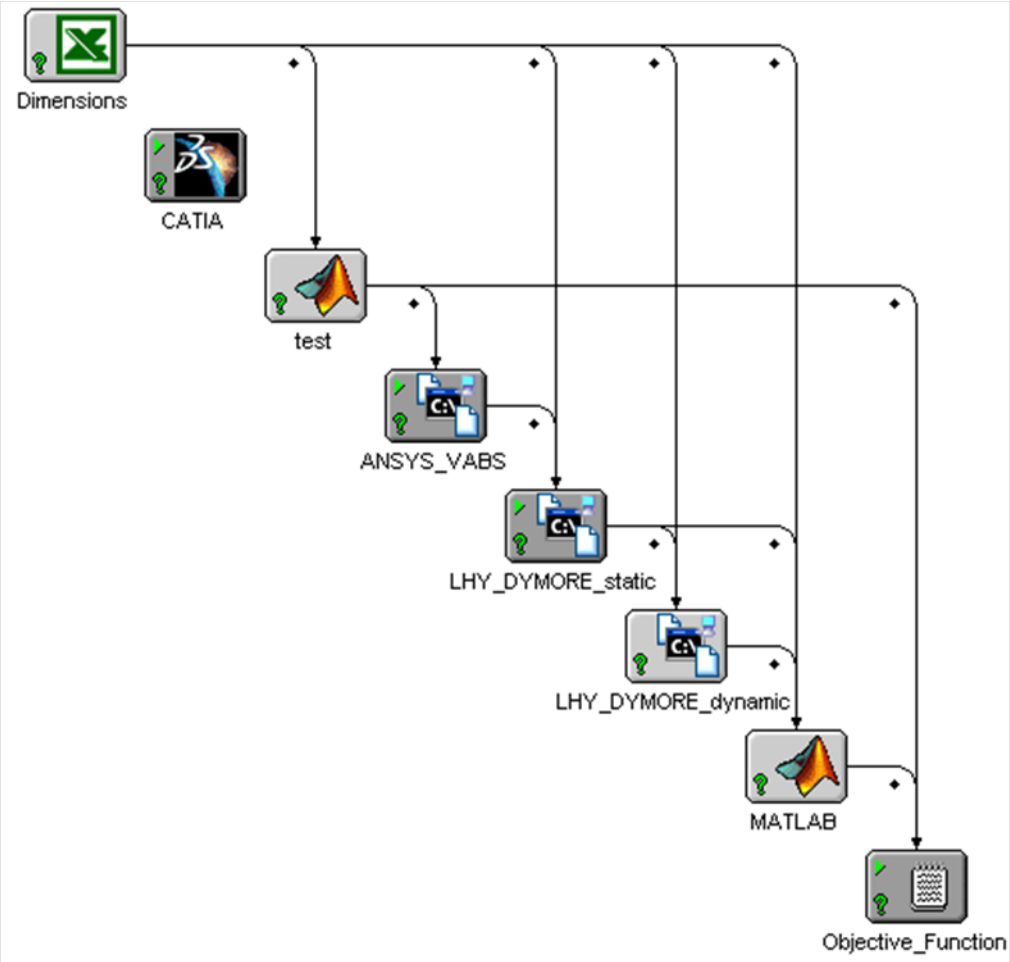


Figure 36: Problem setup in ModelCenter

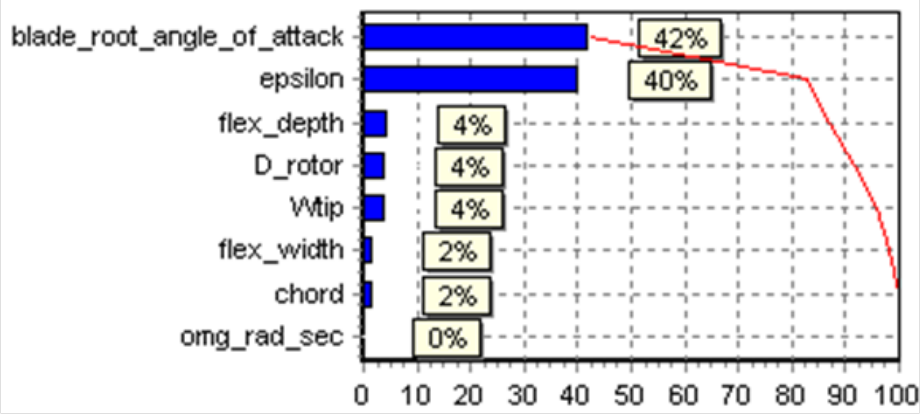


Figure 37: Most effective design variables

Table 4: Initial and final design parameter values for the entire process

Design variable (parameter)	Initial value	Final value	unit
Rotor radius (R)	18	14.17	ft
Rotor operating angular velocity (Ω)	33.33	32.81	rad/s
Blade chord length (c)	11	11.99	in
Blade forward sweep angle (ζ)	2	2.8	deg
Blade linear twist angle (θ_{tw})	12	8.99	deg
Blade tip mass (m_{tip})	0.1001	0.1109	slug
Flexure arm width	0.6	0.452	in
Flexure arm thickness	0.2	0.209	in

taken from the result of the previous step. Other parameters are kept constant at the values found in the same step.

The configuration has not changed much at the end of the line search. The final configuration of the rotor is presented in Table 4. The responses of the rotor at the final configuration are shown in Figures 38, 39, and 40. The forces and moments at the rotor hub are also alleviated, as seen in Figures 41, 42, 43, 44, 45, and 46. The **x**, **y** and **z** axes of the system are shown in Figure 49. Fan plot of the resulting design in vacuum is given in Figure 47.

Auto-trim theory predicts that under aerodynamic forces, the feathering frequency decreases and comes closer to 1P. The effect of aerodynamics can be predicted by analytical formulas. Derivation of those formulas are presented in Appendix C.2. Modifying those calculations by substituting for the values of the optimal configuration, the fan plot with the aerodynamic effects is generated. It is presented in Figure 48. It is clearly seen in the figure that the first feathering frequency approaches 1P as the rotor blade rotates faster.

The CATIA model of the design has been updated automatically during the analyses. Figure 49 shows the final design . The difference between the initial and final flexure configurations is depicted in Figure 50.

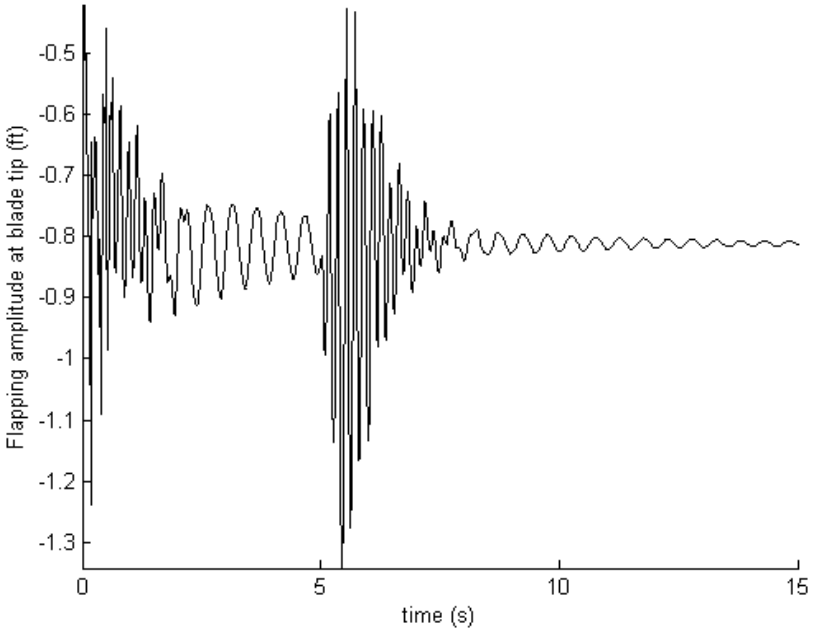


Figure 38: Optimized flapping response of the blade

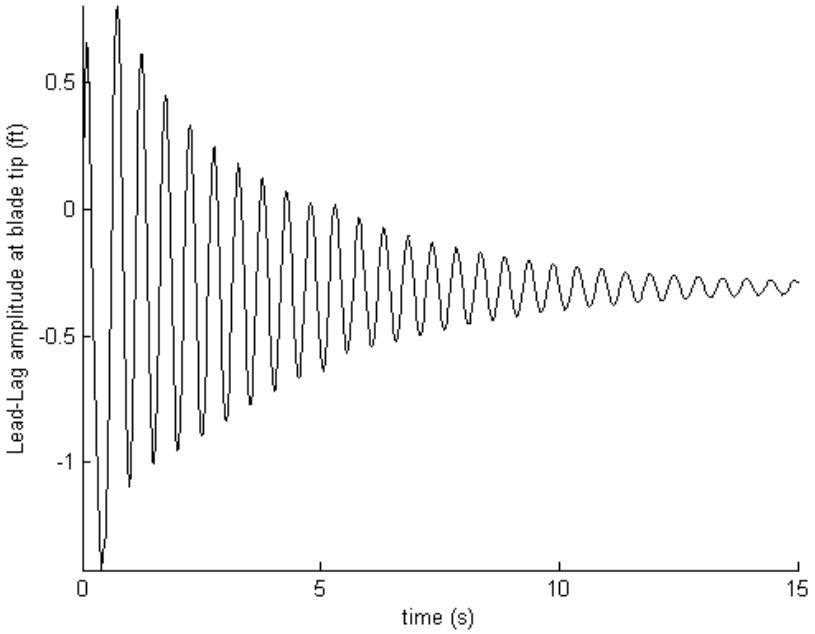


Figure 39: Optimized lead-lag response of the blade

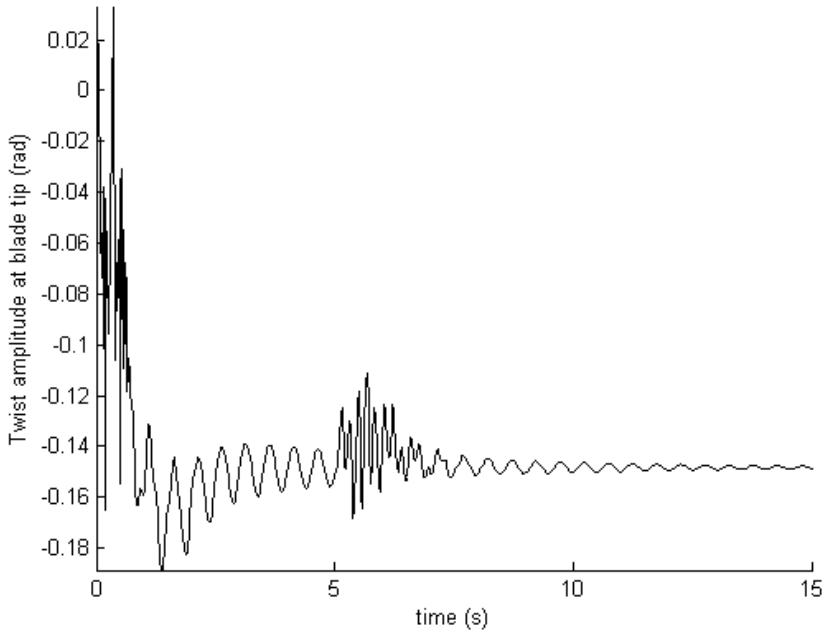


Figure 40: Optimized twisting response of the blade tip

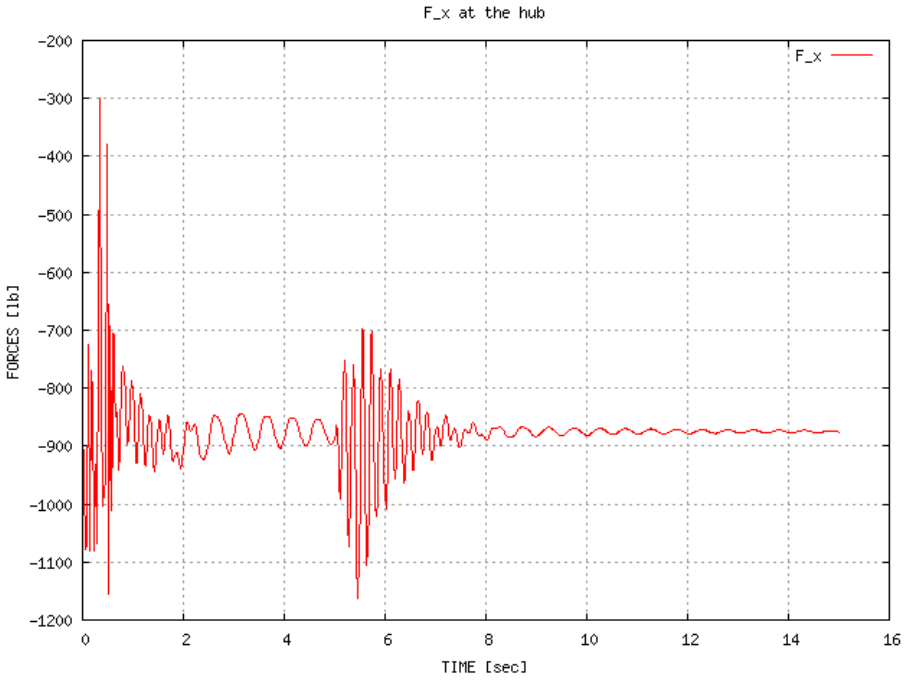


Figure 41: Span-wise forces at the hub

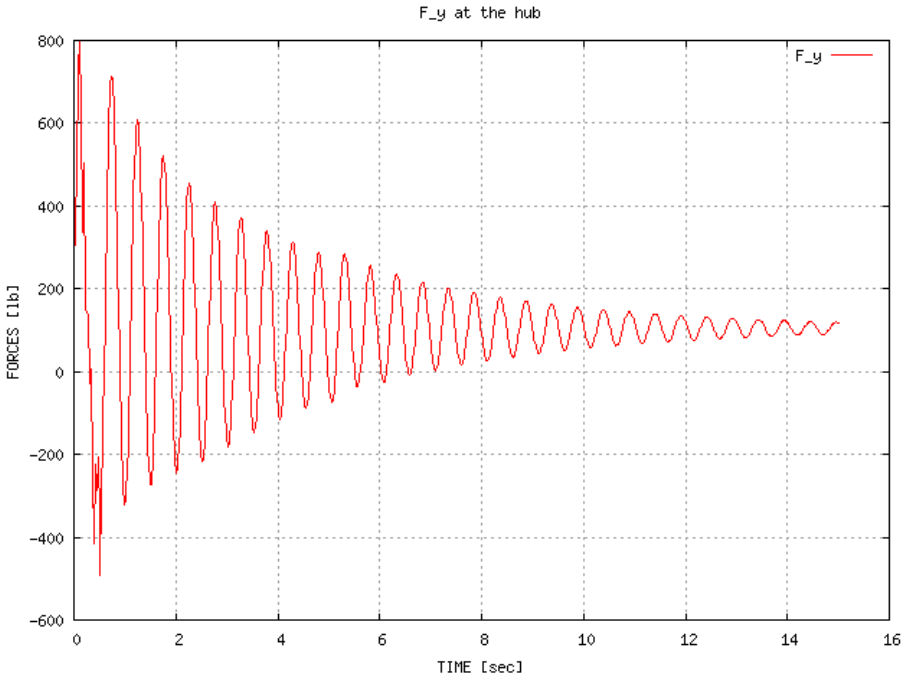


Figure 42: Chord-wise forces at the hub

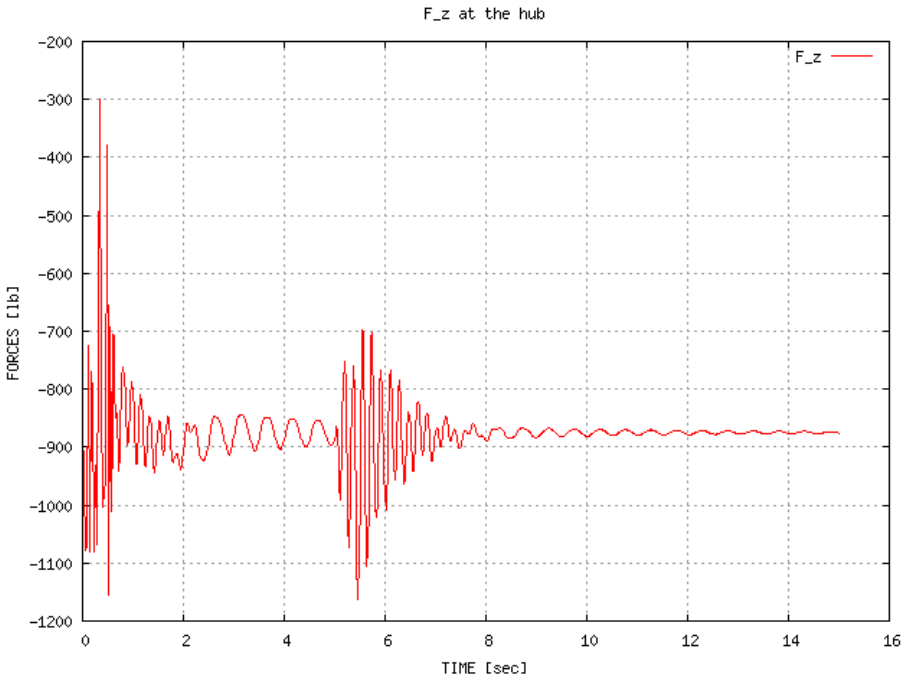


Figure 43: Flap-wise forces at the hub

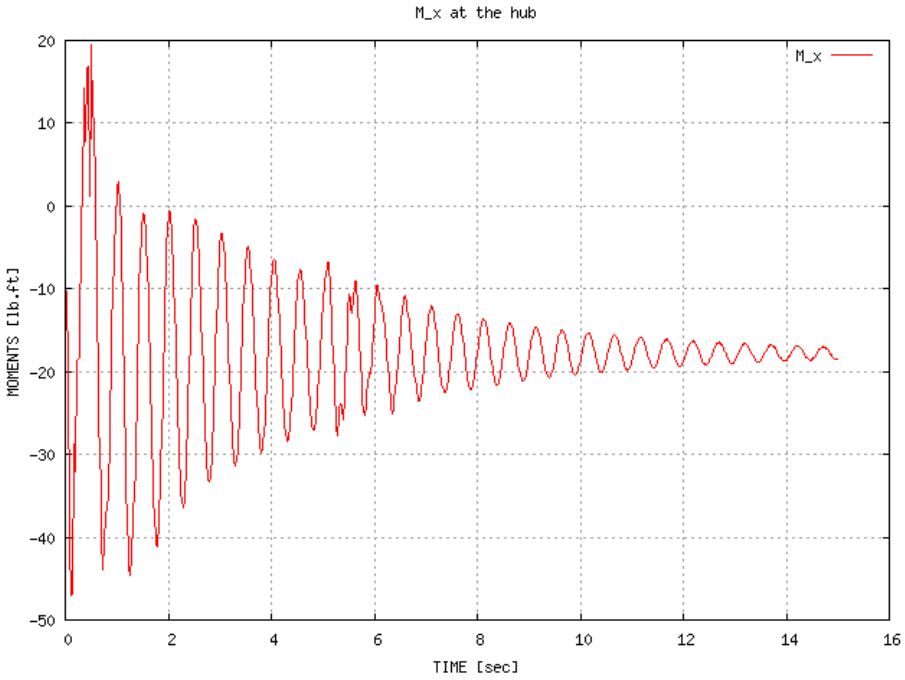


Figure 44: Twisting moments at the hub

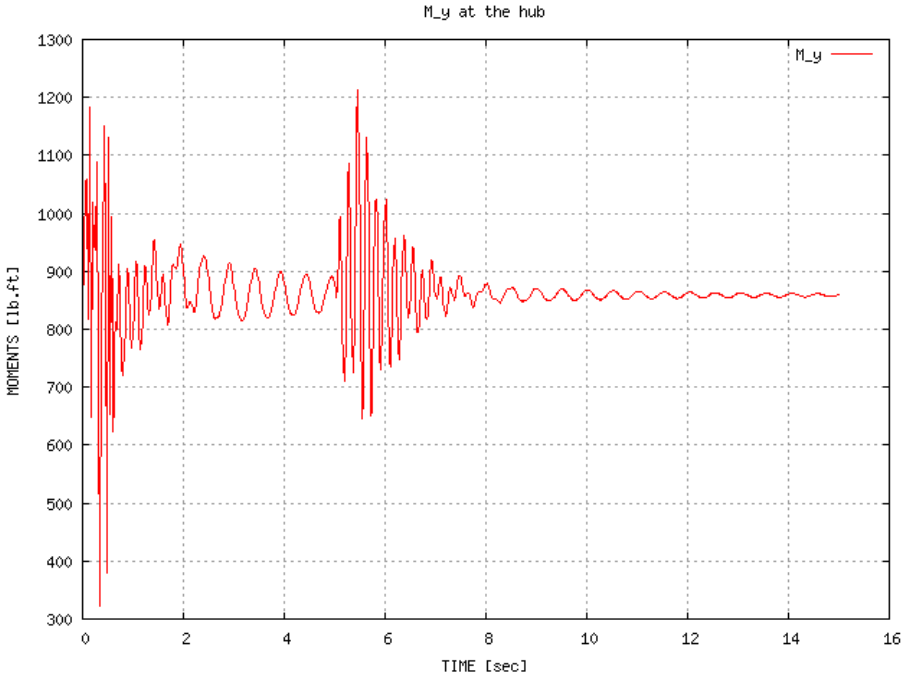


Figure 45: Flapping moments at the hub

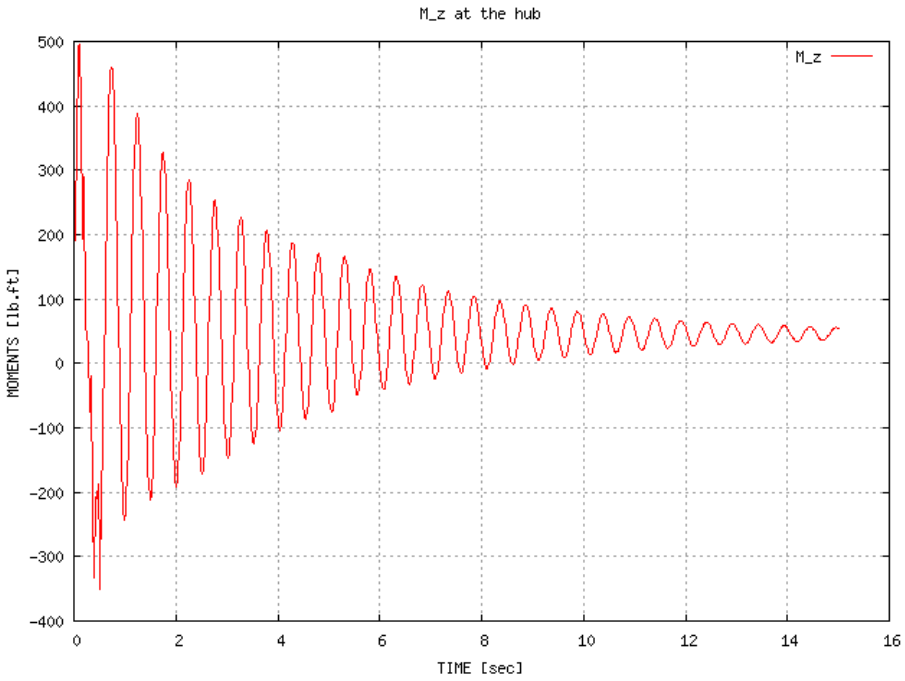


Figure 46: Lead-lag moments at the hub

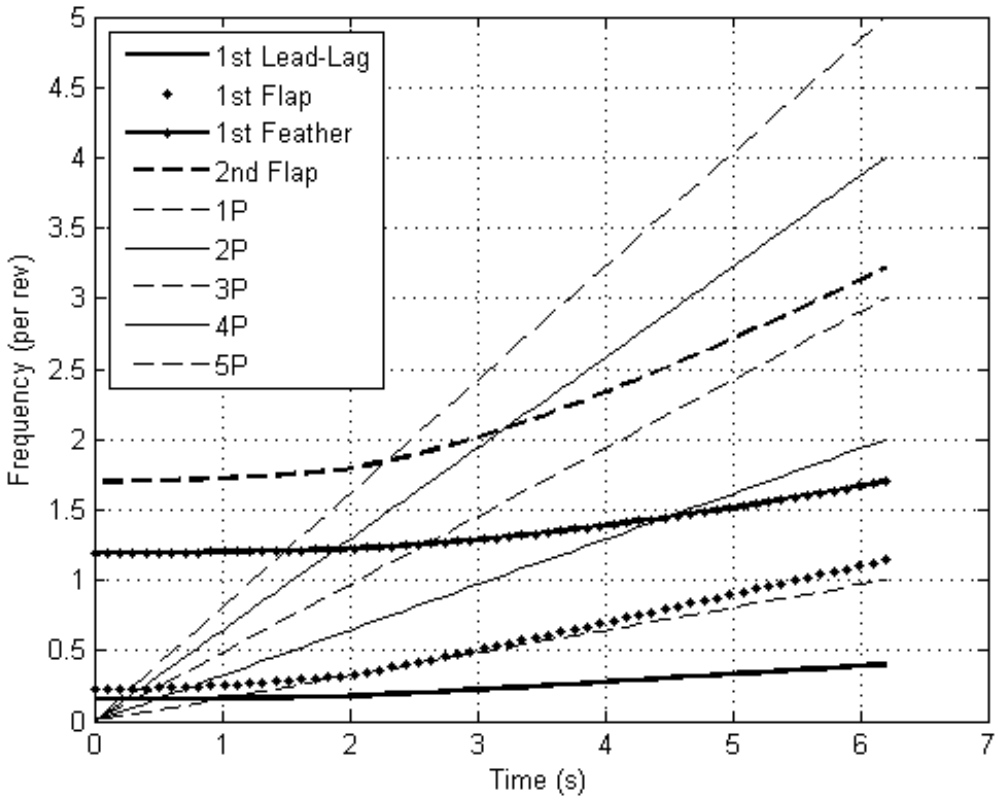


Figure 47: Fan plot of optimum design in vacuum

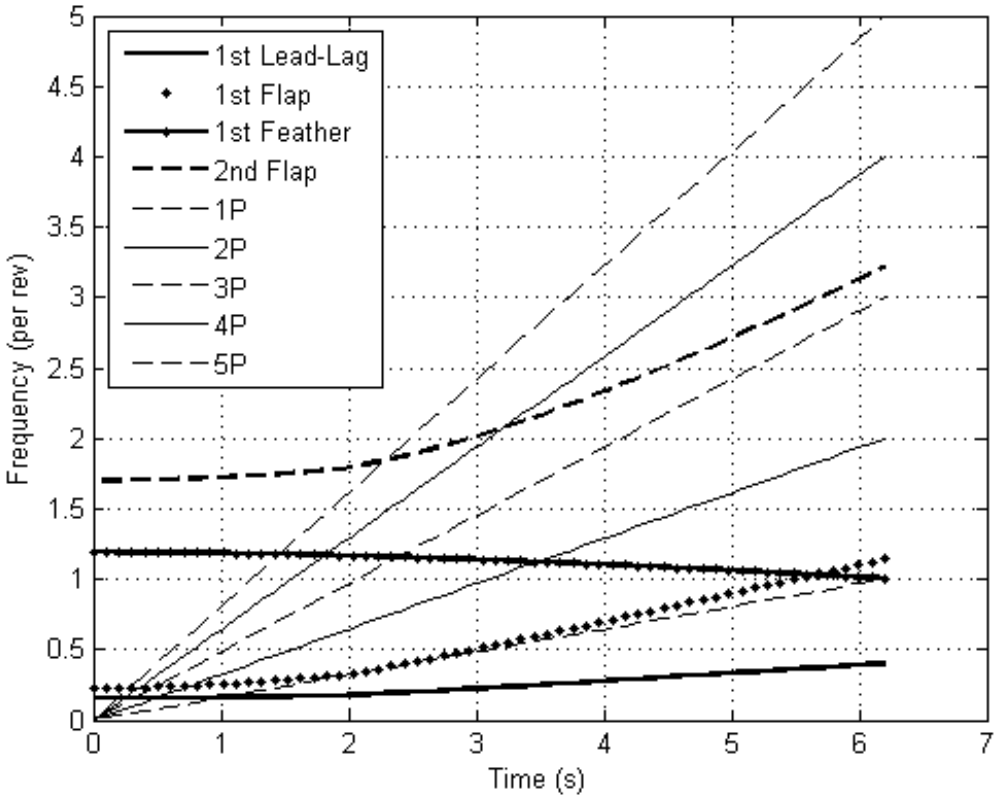


Figure 48: Fan plot of optimum design with the effect of aerodynamics

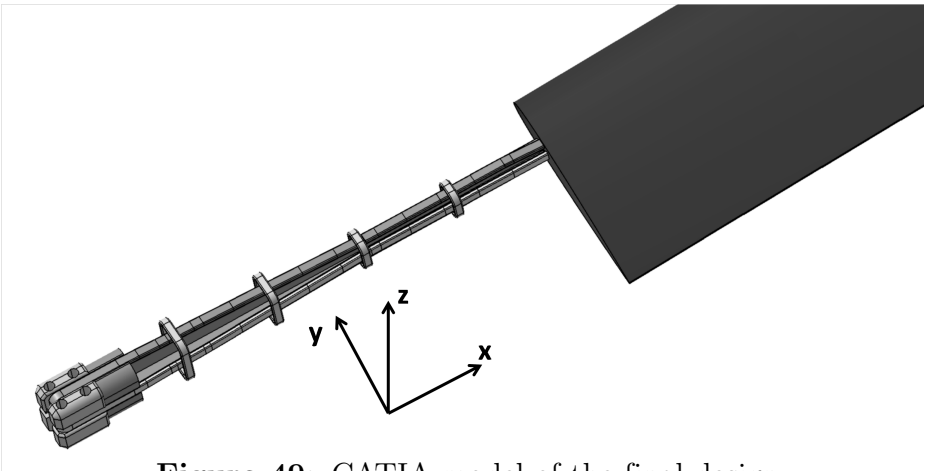


Figure 49: CATIA model of the final design

Figure 50: The initial (top) and final (bottom) flexure configurations

CHAPTER V

RESULTS AND CONCLUSIONS

The resulting design have disturbance responses in Figure 38 noticeably different from the original design, if compared to Figure 30. Hanson's original configuration would still be a good design if an optimization study had not been performed. Nevertheless, effect of the disturbance is alleviated more uniformly in the optimized design. Likewise, leag-lag and pitching responses in Figures 39 and 40 have also been stabilized further. It may therefore be concluded that the optimization methodology proposed in Chapter 3 is effective at finding optimum solutions, provided the objective function is defined properly.

Sweep angle was one of the most important parameters in the first optimization step of identifying key variables. This result is in agreement with the Auto-Trim theory, since it is believed that the Auto-Trim system needs forward sweep for stable operation.

Hanson's design has another unique feature, such that the feathering frequency of the blade approaches the operating RPM with the help of aerodynamic forces and forward sweep of the blade. It is currently not possible to calculate this effect with DYMORE, but it is incorporated into the optimization problem as an analytically calculated constraint. At the end of the analysis, analytical formulas were used to calculate the effects of aerodynamics. The aerodynamic forces clearly reduce the frequency. The first feathering frequency is considerably lower near operating RPM. This result is in complete agreement with the Auto-Trim theory, as it predicts that with forward sweep, the first lead lag frequency approaches 0.4P, and the first flapping and first feathering frequencies approach 1P.

It might be possible to improve the scope of the design further by introducing more design variables such as parameters related to blade structural members, chordwise CG location of the blade, blade taper angle, and new materials.

Estimated times for a manual and automated design are given in tables 5 and 6, respectively. A comparison of those two rough estimates shows the amount of improvement as a result of automation over manual approach. The analyses performed in this study include CAD modeling, structural analysis and dynamic analysis. More detail regarding the analyses is provided in the subsequent chapters. In Table 5, the analysis is assumed to be set up by modeling the system in consideration in each of the software tools. Then it is modified gradually after each run of all the relevant analysis software, in a total of five iterations. The author has not applied the manual approach entirely in this study, thus the number of iterations may differ depending on the particular design and analyses. The manual method was performed only once, without performing any iterations. The number of iterations was assumed based on statistical data: A survey among automotive companies conducted by Dassault Systemes, a CAD software developer based in France, shows that about 50% of all design time is spent on CAD models. When this number is projected in the time calculations in Table 5, the time spent for CAD is approximately half of the total design time when five design iterations are performed.

In Table 5, modification of the CAD and analysis models take slightly less time in each iteration. This is because in each iteration, the designer is assumed to grow more familiar with the design, therefore there is a learning curve effect. He/she spends less time making changes on the model. The formula used in Table 5 is assumed to be $exp(-0.001 n_{iter})$, where n_{iter} is the iteration number.

The total time required for performing the analysis using the traditional way is estimated in Table 5 to be 116.5 hours. This value is highly dependent on the number of iterations, which might be much more than five for complicated systems. The time

required for performing the same analysis in an automated fashion is estimated to be 91 hours, when the iteration times are included. When the author has performed this analysis he recorded several iteration steps performed by the optimizer. The difference of the iterations between manual and automated methods is that, the modifications and software runs are performed entirely by the computer in the automated case. Therefore the execution of the optimizer and analysis codes can be realized during off-business hours. As a result, the time required for iterations may be omitted. Another reason for omitting this time is the fact that the duration of the analysis is highly dependent on the specific computer hardware. A computer with several fast processors will certainly finish the task much earlier than a low-priced regular office computer.

If the iteration time required for the automated analysis is removed from the estimations, the total time required for the automated case reduces to 75 hours. When compared to manual analysis case, this is a 36% reduction in analysis time. In other words, the modeling and engineering man-hours needed for the same study with the automated method is about two thirds of the man-hours needed with the traditional method. The analyses are handled by computers in both cases, but the computation time is longer for the automated case, since remodeling is also performed by computers, rather than engineers. This result favors the automated method over the other one in analyses of similar or further complexity. It is worth noting that this estimation is very conservative because it assumes that only five iterations are required. In reality, however, many more iterations may be necessary for performing a design optimization of this nature. In addition, in case the design goals change after the initial system setup, it is relatively easy to modify the system to conform to the new objectives. For instance, it would take little time to change the objective function from alleviating vibrations, to minimize hub forces. No remodeling in CAD or any other analysis tool is necessary. Such flexibility is another advantage of this

Table 5: Time spent during a traditional design

Design steps	Time required [hrs]	Iteration number 1	2	3	4	5
CAD modeling (not parametric)	20.00					
Structural analysis software modeling and code execution	6.00					
Dynamic analysis software modeling and code execution	16.00					
Reviewing results	1.00					
Modifying CAD model		8.00	7.92	7.84	7.76	7.69
Modifying structural analysis model and code execution		5.00	4.95	4.90	4.85	4.80
Modifying dynamic analysis model and code execution		1.00	0.99	0.98	0.97	0.96
Reviewing results		1.00	0.99	0.98	0.97	0.96
Total time in iteration		15.00	14.85	14.70	14.56	14.41
Total CAD time	59.21					
Initial design time	43.00					
Total design time with iterations	**116.52**					

method over traditional design methods. Flexibility and efficiency achieved in this study results in cost and time savings in the overall design process, thus adjusting the current design methods, the aerospace industry will probably benefit from the paradigm shift discussed in this thesis.

5.1 Recommendations and Future Work

As mentioned in Chapter III, the analysis and optimization method explained in this study can be performed in the preliminary design stage, in order to obtain as much knowledge as possible through analysis earlier in the design. Additionally, if desired, it can be utilized in the detail design stage as well, because it is possible to integrate more design and analysis disciplines, more detailed models, and other objective functions and constraints into the existing system. In the detail design

Table 6: Time spent during automated design

Design steps	Time required [hrs]
CAD modeling (parametric)	26
Structural Analysis Software macro generation for automated modeling	12
Dynamic Analysis software modeling	18
Developing analysis output processing code	8
Setting up communication and optimization software	10
Running communication and optimization software *†	16
Reviewing results	1
Total design time with iterations	91
Design time without iterations	**75**

* Running the optimizer does not require designer attention at all times.
† The duration of the optimization process depends on computer hardware.

stage, however, manufacturing considerations, in addition to product design issues, have to be taken into account. The coupling of manufacturing into product design is realized by the IPPD process, as shown in Figure 8.

The optimization method presented here has already been used by Chae et al.[3] within an IPPD framework for the purpose of product and process optimization of rotorcraft, albeit in a simple way. They employed ModelCenter for implementation of the framework, together with CIRADS as the vehicle sizing tool, CATIA as the CAD tool, DYMORE, ANSYS and VABS for structural/dynamic analysis, GENCAS as the CFD tool, FLIGHTLAB as stability and control tool, and DELMIA as the manufacturing simulation tool. Their dynamic analysis and CAD integration method is the same method as described in this thesis. The output from CIRADS was used as input for CATIA and dynamic analysis, and results of those analyses were easily transferred to other disciplines. This model can be employed in a detailed design setting with more complex and complete models and detailed manufacturing process instructions.

Due to the flexible nature of the analysis system established in this study, the rotor

blade model can be altered or totally reconstructed to accommodate novel features such as blade tip sweep or tabs on the existing model, or distinctive designs such as mission adaptive rotors or reaction drives with tip jets.

Design automation, reducing human interaction, incorporating a parametric CAD tool, integrating a particular analysis into the overall design loop, and in turn, cost reduction by means of improved efficiency are the goals of this study, that will contribute to the state-of-the-art in rotor design. An efficient system design methodology needs all design disciplines to work coherently and communicate with each other during the entire design process. ModelCenter is the software tool utilized for providing an environment in which several analysis and design disciplines can work together, transferring data from one another as necessary. The ModelCenter model in Figure 36 can be placed in a full fledged rotor design loop, such as the one in Figure 20, and then the entire design can be carried out automatically, provided information is transferred between analyses without human interaction. In the current problem, sectional properties of the flexure beams are modeled in ANSYS via macros automatically. Then DYMORE input files are generated and optimization is performed without human interaction. At the end of the optimization, the new design information is sent back to CATIA, the CAD tool, to modify geometry accordingly. In a fully united design system, the updated geometry may then be fed forward to next analysis discipline. In this study, it is used for another iteration during the optimization. In conclusion, entire preliminary design loop may be completed in the fashion described above. The case of blade optimization considered here demonstrates that it is possible to wrap several analysis software together in order to perform an automated optimization, even for a discipline with relatively intricate relations.

APPENDIX A

DESCRIPTION OF COMMERCIAL SOFTWARE TOOLS

A.1 CATIA

CATIA (Computer Aided Three Dimensional Interactive Application) is part of a multi-platform CAD/CAM/CAE commercial software suite developed by French company: Dassault Systemes. CATIA is widely used throughout the engineering industry for modeling mechanical and electrical parts and assemblies, especially in the automotive and aerospace sectors. The software is capable of handling complex assemblies composed of a great number of parts.

CATIA is suitable for modeling intricate part geometry in desired precision, and creating blueprints for manufacturing. Higher precision can be achieved if all the necessary lengths and angles of the features in a part are well defined. The CATIA models are parametric, that is, the entire model is updated automatically when any geometry is modified.

It is possible to define material properties for solid models. When materials are assigned to all parts, it is possible to obtain total weight, CG location, and moments of inertia for individual parts, or the entire product. For subsequent computer simulations and structural and aerodynamic analysis, existing parts or products may be meshed using numerous surface and solid mesh elements.

Operation of mechanical systems such as gearbox assemblies can be demonstrated by means of animating the system at desired speeds. Also for demonstration and marketing purposes, all parts and assemblies may be realistically rendered with various artistic lighting effects.

CATIA includes accurate human models for design considerations regarding ergonomics. Various human sizes ranging from 5-percentile Asian female to 95-percentile American male may be utilized to discover pilot/driver field of view inside a vehicle, and optimum position of control input devices.

APPENDIX B

DESCRIPTION OF IN-HOUSE ANALYSIS TOOLS

B.1 VABS

VABS is a general-purpose, finite-element based beam cross-sectional analysis code based on the Variational Asymptotic Beam Sectional analysis that is developed using a generalized Vlasov theory for composite beams with arbitrary geometric and material sectional properties. This tool rigorously splits the geometrically-nonlinear, three-dimensional elasticity problem into a linear, two-dimensional, cross-sectional analysis and a nonlinear, one-dimensional, beam analysis. Its schematic organization is depicted in Figure 51.

VABS is a 2-D in-house finite element code, originating as far back as 1992 [71]. A new generation of VABS was developed over the initial code by adding the features such as calculating the principal bending axes and the corresponding principal moments of inertia, calculating the neutral axes, eliminating the point constraints, solving the rank deficient linear system exactly and speed improvement [72, 73]. This modification resulted in at least 20% improvement in speed compared to the first generation of VABS.

VABS implements the various beam theories [72, 74, 75, 76] using the finite element method-based on the variational asymptotic method. It uses triangular or quadrilateral elements with 3, 4, 5, 6 nodes or 4, 5, 6, 7, 8, 9 nodes, however, if VABS-ANSYS macro is to be used, 3 or 4 noded triangular elements or 3 or 8 noded quadrilateral elements can be selected. VABS generates 6x6 cross-sectional mass matrix, 4x4 and 6x6 stiffness matrix based on classical model and Timoshenko model.

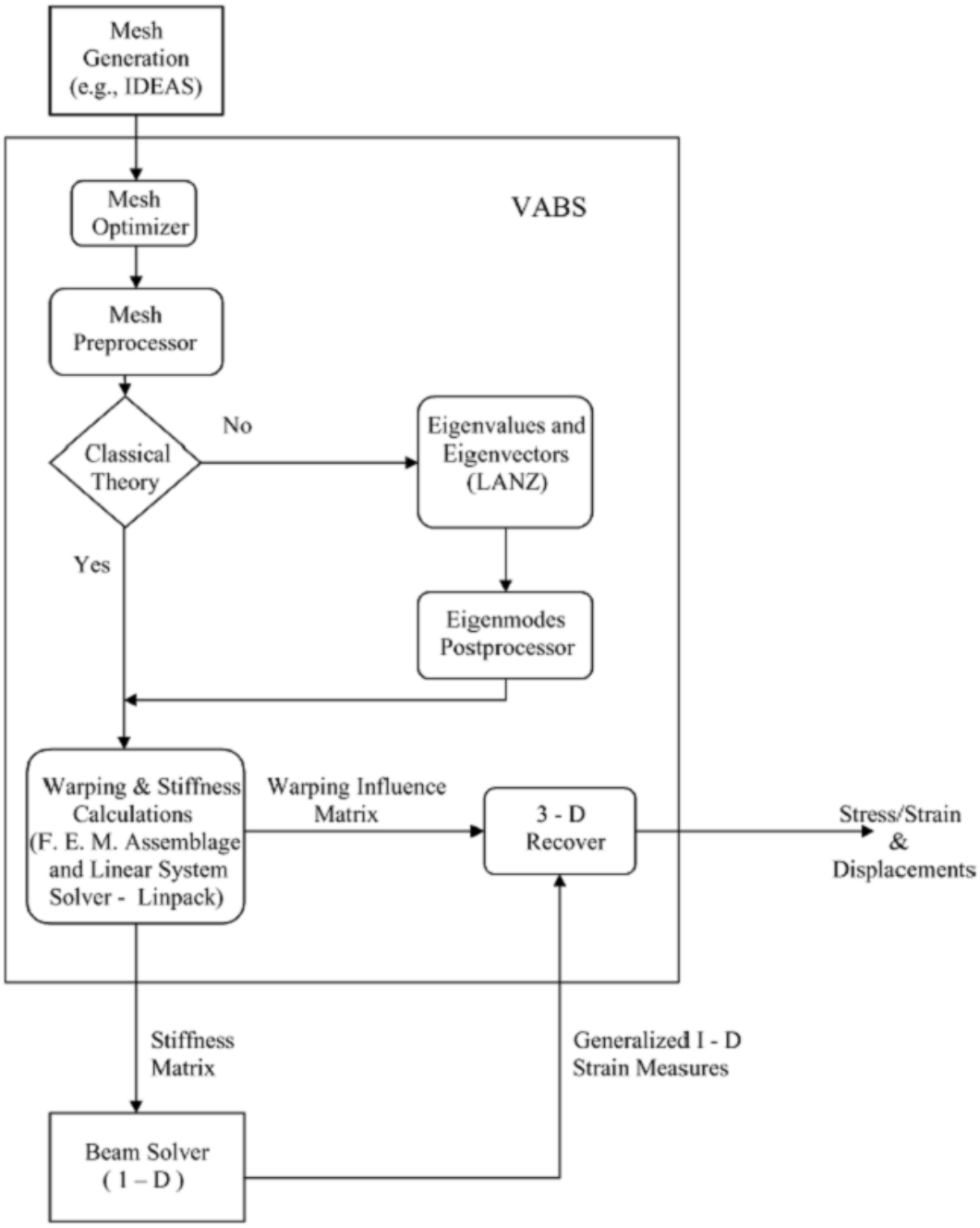


Figure 51: Schematic organization of VABS

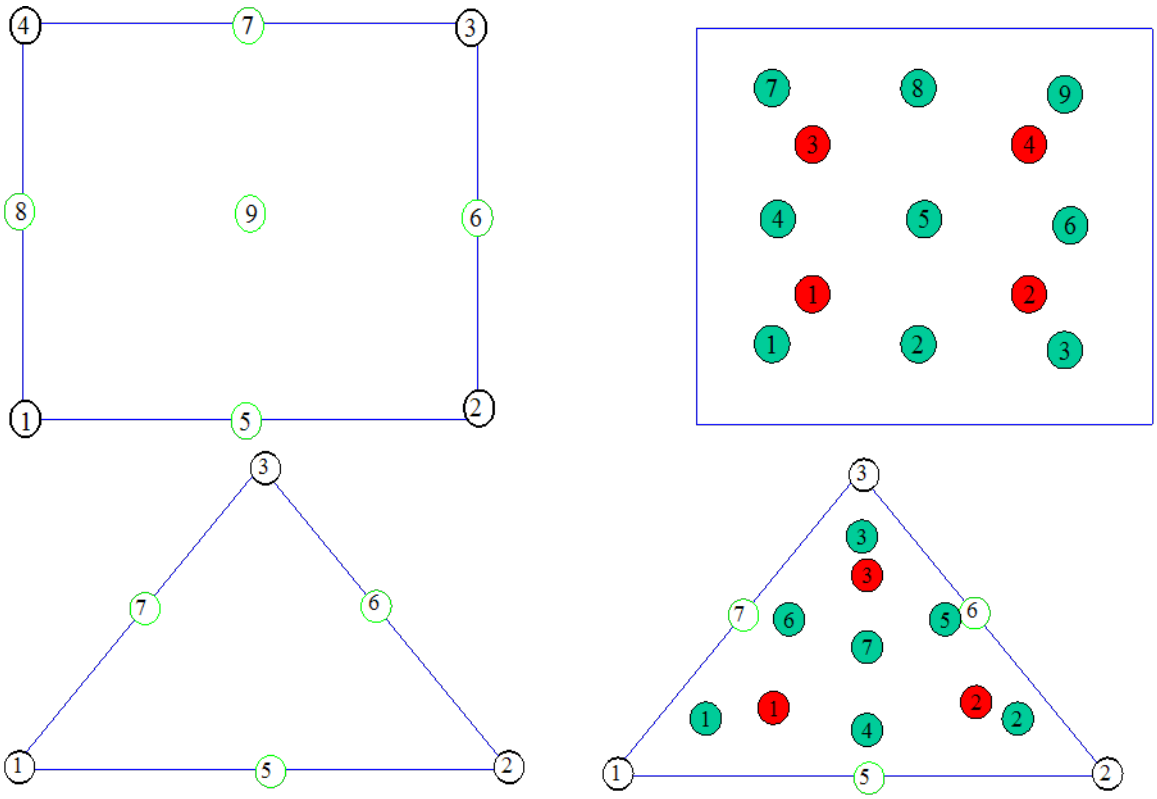


Figure 52: Element types in VABS. Upper two rectangles: Quadrilateral elements. Lower two triangles: Triangular elements

These beam models can be prismatic or initially twisted. Also, VABS result file contains mass center and shear center. The element types used in VABS are shown in Figure 52 [77].

Gaussian integration schemes are utilized to recover 3-D stresses and strains. The red numbers represents Gaussian points for linear elements and the green represents Gaussian points for quadratic elements.

Sign convention of lay-up angles in VABS is depicted in Figure 53 [77]. The right-hand coordinate system is adopted. As shown in the figure, x_1 is along the beam axis and x_2 and x_3 are the local Cartesian coordinates of the cross section. The coordinate system (x_1, x_2, x_3) is a global system used to define the geometry, (e_1, e_2, e_3) used by to define the material properties, and (y_1, y_2, y_3) is used to define the ply plane.

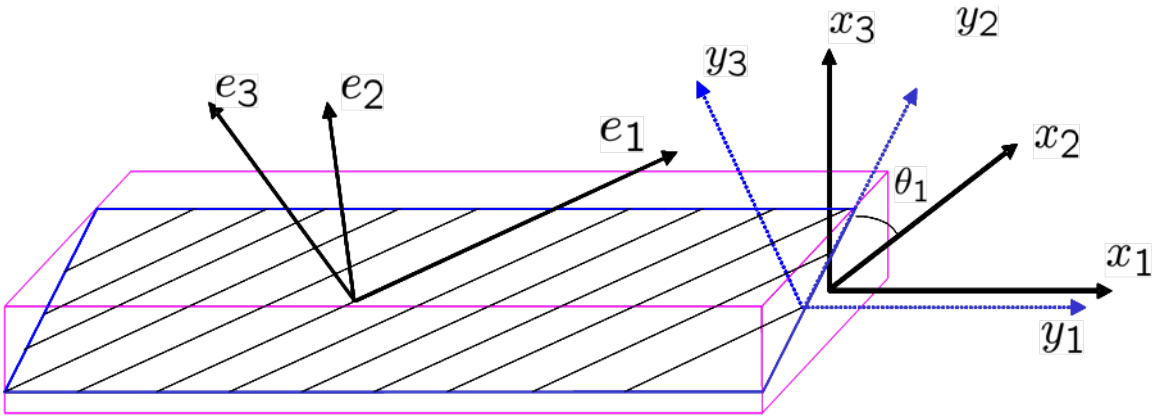


Figure 53: Orientation of ply plane (y_1, y_2, y_3) in global reference system (x_1, x_2, x_3)

B.2 DYMORE

DYMORE is a finite element based nonlinear elastic multibody dynamic analysis tool developed by Professor Olivier Bauchau and colleagues in Georgia Institute of Technology. This tool provides a level of generality and flexibility required to solve complex problems such as rotorcraft analysis.

The multibody dynamic analysis is cast within the framework of nonlinear finite element methods, and the element library includes rigid and deformable bodies as well as joint elements. Deformable bodies are modeled with the finite element method. Brief description of the elements library implemented in DYMORE is given as follows.

Rigid Bodies: Rigid components used for components whose flexibility can be neglected or for introducing localized masses.

Cables: 1-D flexible components whose bending stiffness can be neglected.

Beams: 1-D flexible components used for modeling slender, flexible members. Both straight and curved beams can be modeled. Components made of laminated composite materials can be modeled. It is possible to represent shearing deformation effects, the offset of the center of mass and of the shear center from the beam reference line, and all the elastic couplings that can arise from the use of tailored composite materials.

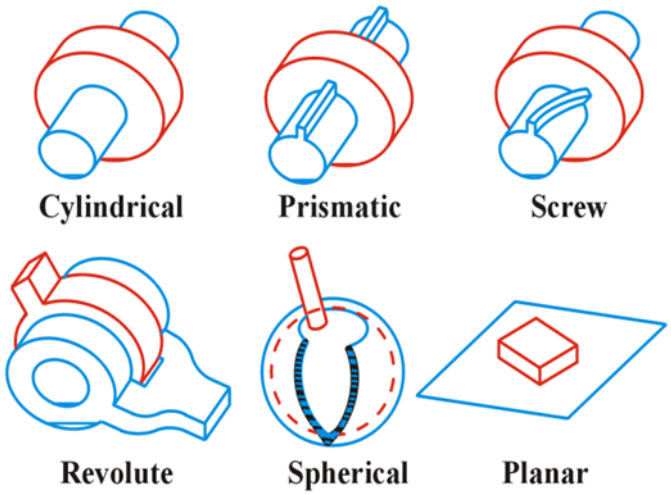


Figure 54: Various joints in DYMORE (the six lower pairs)

Shells: 2-D components used for modeling thin, flexible members. Both flat plates and curved shells can be modeled. Components made of laminated composite materials can be modeled. It is possible to represent shearing deformation effects, the offset of the center of mass from the shell reference line, and all the elastic couplings that can arise from the use of tailored composite materials.

Lower Pairs of Joints: Most joints used for practical applications can be modeled in terms of the so called lower pairs: the revolute, prismatic, screw, cylindrical, planar and spherical joints. These joints are shown in Figure 54 [78].

Sliding Joints: Constraint elements that involves kinematic constraints at the instantaneous point of contact between the sliding bodies.

Planar Contact Joints: Constraint elements to model the intermittent contact between components of the model. Coulomb's friction law has been extensively used to model friction forces.

Backlash Joints: Backlash behavior can be added to the modeling of revolute joints. The joint is generally free to rotate, but when the relative rotation reaches a preset value, a unilateral contact condition is activated corresponding to the backlash stop.

Mathematical model of these elements are provided in Theory Manual [78] and also numerous publications by Bauchau and others [79, 80, 81, 82, 83, 84, 85, 86, 87].

Simplified models based on lifting line theory and vortex wake models, or sophisticated computational fluid dynamics codes can be used for modeling the aerodynamics loads that might be acting on the multibody system. At each time step of the simulation, the aerodynamic loads acting on the system are computed based on the present configuration, and are then used to evaluate the dynamic response. It is also possible to couple the multibody dynamics procedure with a computational fluid dynamics code.

APPENDIX C

THEORY AND BASIC ANALYSES ON HANSON'S AUTO-TRIM ROTOR SYSTEM

C.1 VABS and DYMORE Analyses

The dynamic analysis of the presented rotor system is carried out by using DYMORE. The rotor system is modeled to investigate the natural frequencies and perturbation response behavior. The flexure and blade are modeled by beam elements. Sectional properties are accurately computed using VABS and then inserted into the DYMORE model. Fan plots in vacuum and air are computed to investigate the placement of the natural frequencies to check requirements of the auto trim system.

The rotor system model has four blade assemblies but only one blade assembly is modeled studied for convenience. The various mechanical components of the system are associated with the elements found in the library of DYMORE. Rigid body elements are used to model the shaft and the hub. These rigid bodies are connected via joint elements such as revolute, spherical, sliding and universal joints. The shaft is clamped to the ground by a revolute joint. The DYMORE model of connections is given in Figure 55 on the following page.

The rotor blade and flexure are modeled as beams with heterogeneous cross sections. They are modeled as elastic cantilever beams undergoing flap and lag bending, elastic torsion and axial deformations.In order to compute the equivalent cross sectional stiffness at arbitrary spanwise locations, VABS was employed to obtain acceptable results for the diagonal terms in the stiffness matrix. Especially for sections with isotropic materials, VABS is capable of accurately computing all stiffness terms, including the off-diagonal "coupling" terms, especially for beams constructed using

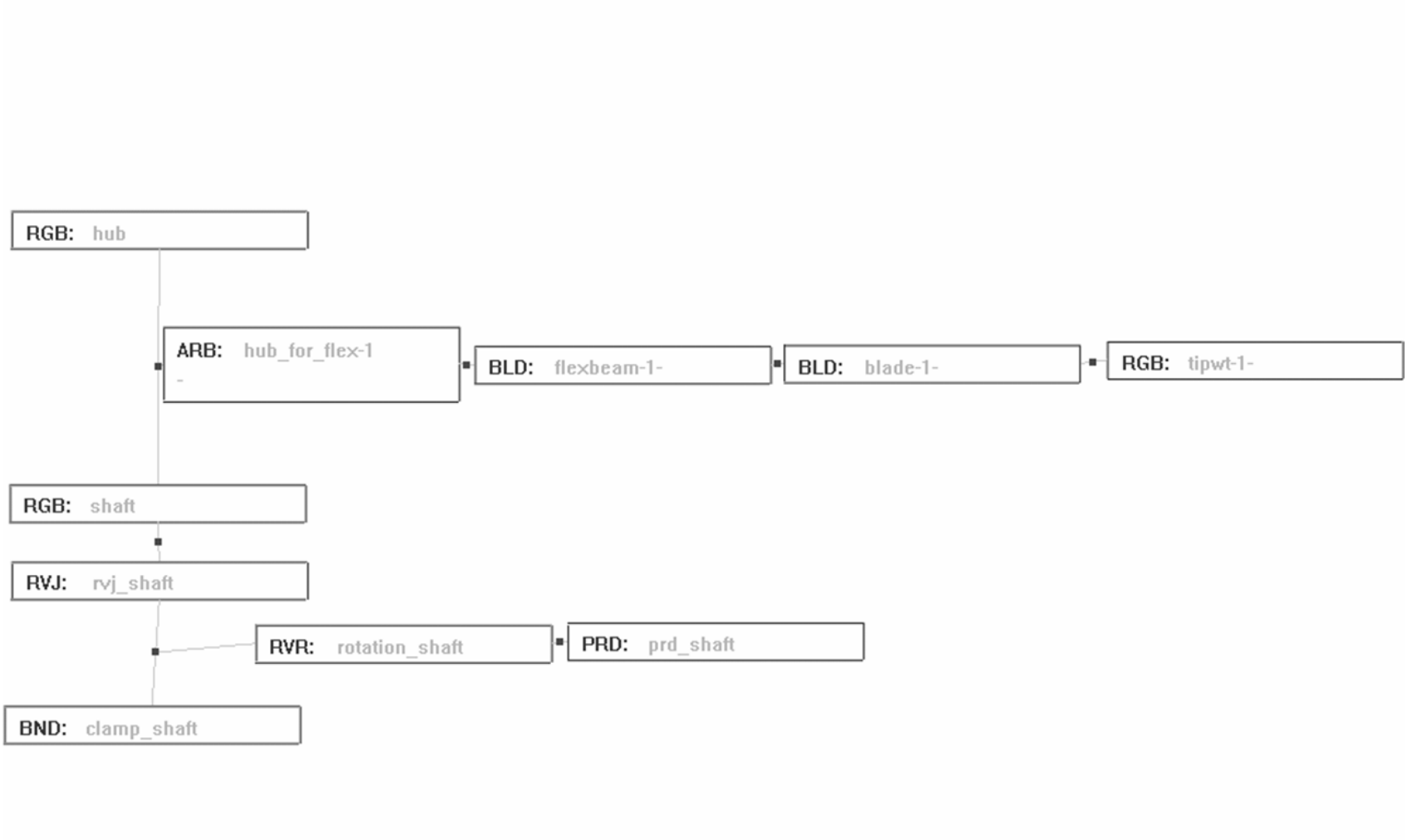


Figure 55: DYMORE model of the rotor system

anisotropic materials such as composites.

The CAD drawing of the entire blade was provided by Hasan İbacoglu at İstanbul Technical University (ITU) for analysis as shown in Figure 56. The CAD drawing was converted to the CATIA model first, and all geometric properties of various blade and flexure cross sections were extracted from that model. Those 2D cross sections are depicted in Figures 57 and 58. The crossection geometries were then imported to ANSYS for meshing. The resulting cross sectional properties were obtained by meshing all the cross sections in ANSYS. The Shell63 element with triangular shape was used in meshing process. The element size was set as 0.001 for auto meshing. The last cross section on the blade was also meshed using same element, but smaller size (0.0005) was used for the auto meshing. Results were reasonably close to the case of size 0.001 mesh output.

Table 7 shows the material properties used for VABS to obtain sectional properties. The values in Table 7 were taken from an earlier report: Dynamic Analysis of the ITU Light Commercial Helicopter (LCH) Main Rotor Blades, prepared by İlyas Toprak. The values of shear moduli (G) for brass, syntactic epoxy foam (SEF) and composite materials were obtained by substituting the value of elasticity modulus (E) into the appropriate equation.

Locations of cross sections on the blade are depicted in Figure 59. There are six cross sections in the flexure portion and six cross sections in the blade portion. The cross sections and their relative distances to their particular reference lines are also shown in Figure 59. Dimensions are in meters. The leftmost line coincides with hub plane and centerline. The reference line of blade cross sections is the first line on the blade. The first blade cross section coincides with the reference line. The distance of this reference line to the origin is 0.98m. The resulting sectional properties such as 6x6 stiffness matrix, 6x6 mass matrix, center of gravity location, and shear center location for each cross sections are shown in Figure 60.

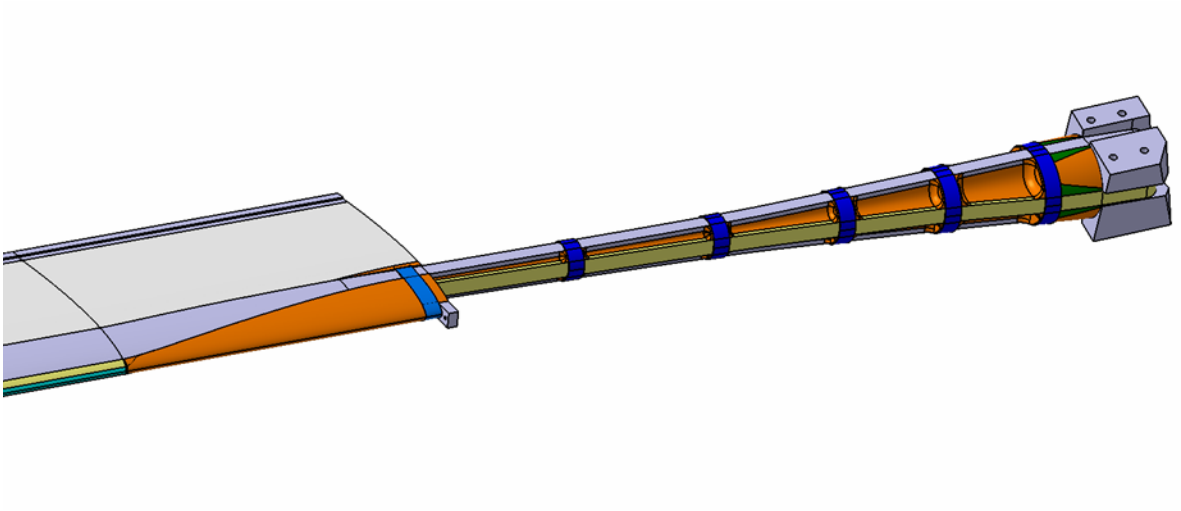


Figure 56: CAD drawing converted to CATIA file

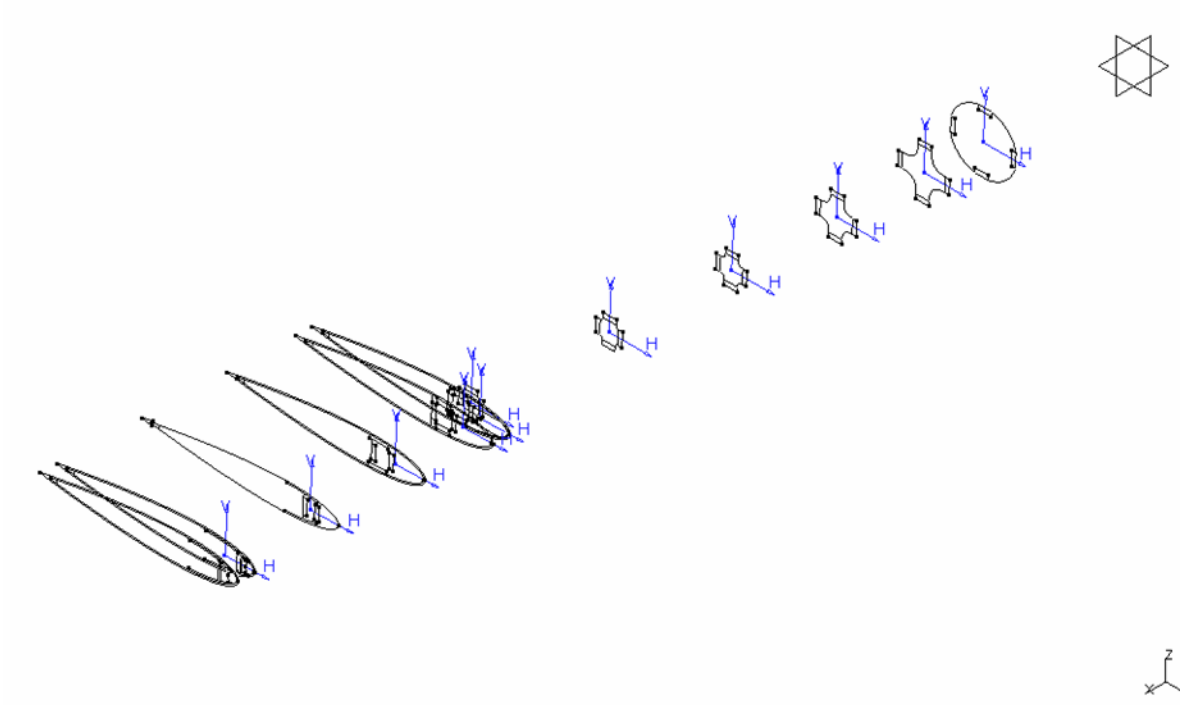


Figure 57: 2D cross sections

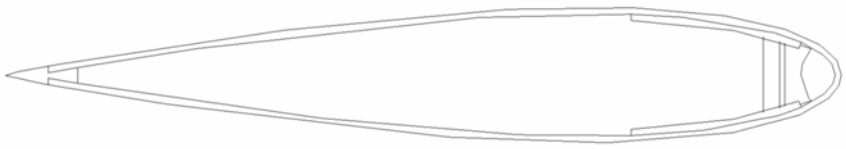


Figure 58: A blade crosssection

Table 7: Material properties for the blade section

	Generic S-Glass / Epoxy Unidirectional Prepreg	**7781/5245 C Glass / Epoxy Fabric Prepreg**	**Unit**
E11	43000	29700	MPa
E22	8900	29700	MPa
G12	4500	5300	MPa
G13	4500	5300	MPa
G23	4500	5300	MPa
Poisson's Ratio	0.27	0.17	
Density	1000	1000	kg/m^3

	Brass	**Foam**	**SEF (Epoxy-fiber)**	**Unit**
E	115000	76	2200	MPa
G	43893.129	29.23076	846.154	MPa
Poisson ratio	0.31	0.3	0.3	
Density	9000	32	1000	kg/m^3

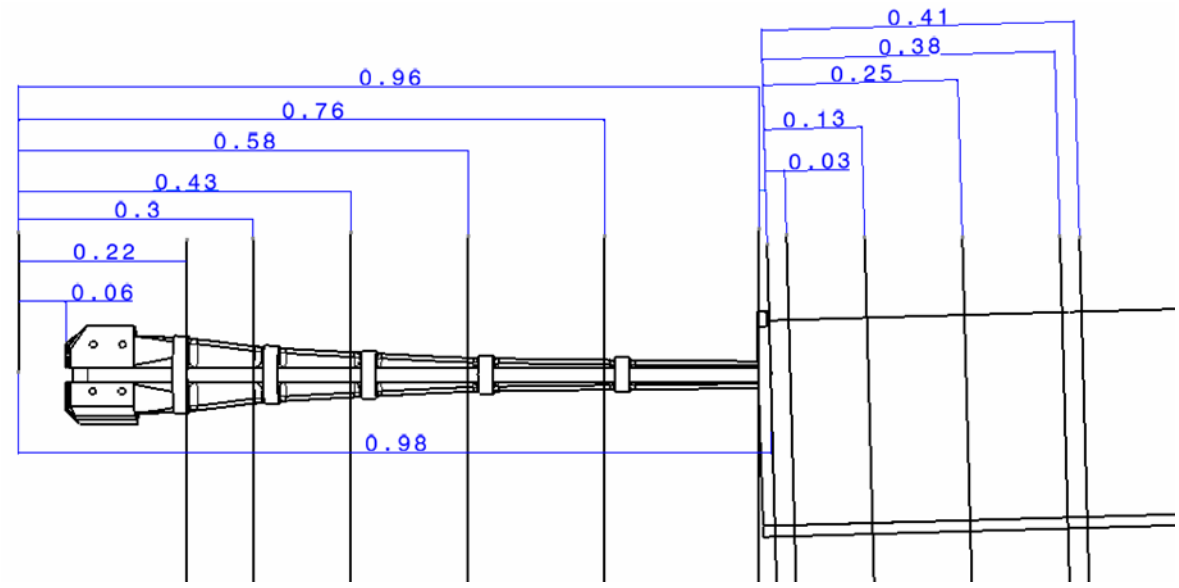


Figure 59: Cross sections of the flexure and blade used in VABS analysis (dimensions are in meters)

```
*****************************************************
*   VARIATIONAL ASYMPTOTIC BEAM SECTIONAL ANALYSIS     *
*                                                      *
*            SCHOOL OF AEROSPACE ENGINEERING           *
*            GEORGIA INSTITUTE OF TECHNOLOGY           *
*****************************************************

THE 6X6 MASS MATRIX (ACCORDING TO DYMORE CONVENTION)
====================================================

       0.199823D+01         0.000000D+00         0.000000D+00         0.000000D+00        -0.535964D-04         0.629003D-01
       0.000000D+00         0.199823D+01         0.000000D+00         0.535964D-04         0.000000D+00         0.000000D+00
       0.000000D+00         0.000000D+00         0.199823D+01        -0.629003D-01         0.000000D+00         0.000000D+00
       0.000000D+00         0.535964D-04        -0.629003D-01         0.165239D-01         0.000000D+00         0.000000D+00
      -0.535964D-04         0.000000D+00         0.000000D+00         0.000000D+00         0.321850D-03        -0.136078D-04
       0.629003D-01         0.000000D+00         0.000000D+00         0.000000D+00        -0.136078D-04         0.162021D-01

THE MASS CENTER
===============================================

 Xm2 =  -3.147806102106772E-002
 Xm3 =  -2.682201688906795E-005
THE 6X6 MASS MATRIX (ACCORDING TO DYMORE CONVENTION)
WITH RESPECT TO A PARALLEL COORDINATE AT THE MASS CENTER
===============================================

       0.199823D+01         0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00
       0.000000D+00         0.199823D+01         0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00
       0.000000D+00         0.000000D+00         0.199823D+01         0.000000D+00         0.000000D+00         0.000000D+00
       0.000000D+00         0.000000D+00         0.000000D+00         0.165239D-01         0.000000D+00         0.000000D+00
       0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00         0.321849D-03        -0.119207D-04
       0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00        -0.119207D-04         0.142221D-01


THE 6X6 MASS MATRIX (ACCORDING TO DYMORE CONVENTION)
WITH RESPECT TO PRINCIPAL INERTIAL AXES WITH ORIGIN AT THE MASS CENTER
===============================================

       0.199823D+01         0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00
       0.000000D+00         0.199823D+01         0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00
       0.000000D+00         0.000000D+00         0.199823D+01         0.000000D+00         0.000000D+00         0.000000D+00
       0.000000D+00         0.000000D+00         0.000000D+00         0.165239D-01         0.000000D+00         0.000000D+00
       0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00         0.321839D-03         0.000000D+00
       0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00         0.000000D+00         0.142221D-01

===============================================

THE PRINCIPAL INERTIAL AXES IS ROTATED FROM THE USER COORDINATES BY
 4.913604994524335E-002

THE CENTROID OF THE CROSS SECTION (PURELY GEOMETRIC)
===============================================

 X2 =  -4.731588331464038E-002
 X3 =  -4.062353118319651E-005
STIFFNESS - WITHOUT CORRECTION
===============================================
      1-extension; 2-twist; 3- and 4-bending

      0.40424821E+08      0.58351118E-09     -0.15979509E+02      0.23394425E+06
      0.58351118E-09      0.11329835E+05     -0.92638116E-10      0.11404661E-09
     -0.15979509E+02     -0.92638116E-10      0.66831681E+04     -0.18222057E+03
      0.23394425E+06      0.11404661E-09     -0.18222057E+03      0.22418004E+06


THE NEUTRAL AXES
===============================================

 Xe2 =  -5.787143797986134E-003
 Xe3 =  -3.952895418699868E-007
TIMOSHENKO STIFFNESS MATRIX
===============================================
    1-EXTENSION;  2,3-SHEAR;  4-TORSION;  5,6-BENDING

      0.40424821E+08      0.13889841E-06      0.12900635E-06      0.17033537E-08     -0.15979509E+02      0.23394425E+06
      0.13889841E-06      0.11541981E+08      0.18300128E+06      0.92222378E+04      0.49386075E-07     -0.46290613E-08
      0.12900635E-06      0.18300128E+06      0.10383709E+07      0.83844733E+04      0.16157703E-08      0.94732289E-08
      0.17033537E-08      0.92222378E+04      0.83844733E+04      0.11402748E+05     -0.46552478E-10      0.18630139E-09
     -0.15979509E+02      0.49386075E-07      0.16157703E-08     -0.46552478E-10      0.66831681E+04     -0.18222057E+03
      0.23394425E+06     -0.46290613E-08      0.94732289E-08      0.18630139E-09     -0.18222057E+03      0.22418004E+06


THE SHEAR CENTER OF THE CROSS SECTION
===============================================
 E2 =   7.956055719812520E-003
 E3 =  -6.728714550528598E-004
```

Figure 60: VABS output for a blade crossection

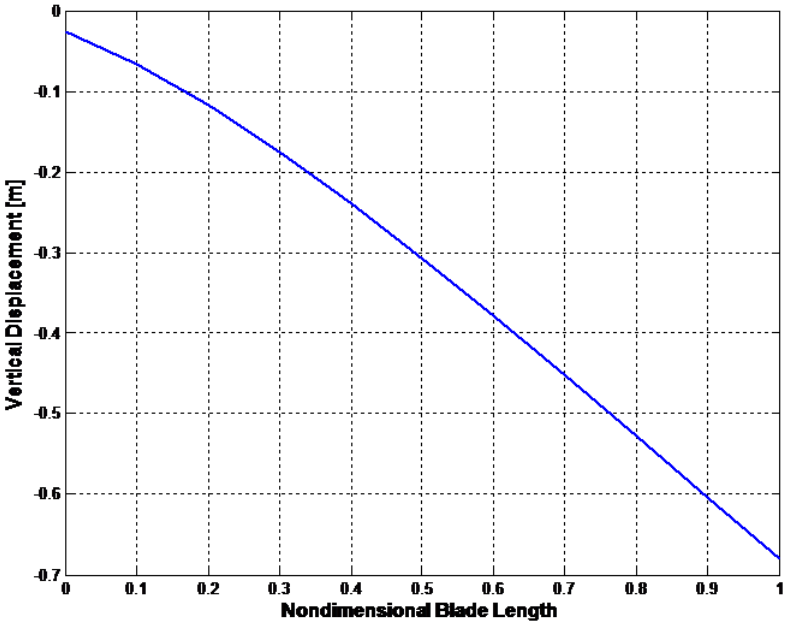


Figure 61: Static droop distance

C.1.1 Analysis Results

The static droop is computed and Figure 61 shows the static droop distances along the blade length. The downward displacement at the blade tip is -0.68m and the initial upward displacement due to the 3.5° precone angle is 0.335m. The net downward displacement below the hub center becomes -0.345m. The blade tip does not sag below the blade root too much, thus a droop-stop mechanism is not necessary.

Vibrations in a helicopter are dependent on good separation from the natural frequencies of the system, especially the forcing functions at 3P, 4P, and 5P for a four-bladed system. The natural frequencies with respect to the rotor speed are of prime importance to the Auto-Trim rotor system.

The possibility of air resonance can usually be eliminated if the first lead-lag frequency is below 0.5P [69]. In order to minimize cyclic control forces, the feathering natural frequency in air needs to be placed near 1P and the feathering frequency needs to be matched with the first flapping frequency for the Auto-Trim system. Hence, the flexure is designed to place the lead-lag and flapping frequency near 0.4P and

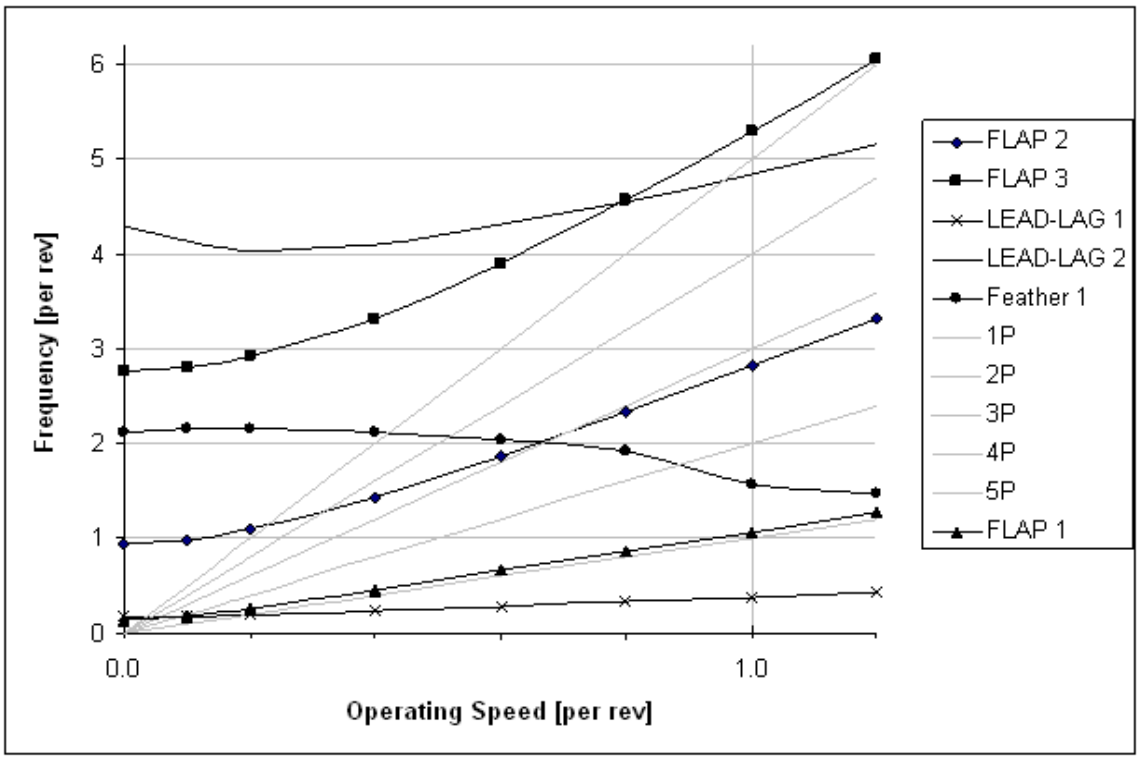


Figure 62: Fan plots in air

1.0P respectively. To ensure the right placement of natural frequencies, the natural frequency map (fan plot) needs to be carefully examined. The fan plots for the model were generated running DYMORE in static analysis mode. Result is shown in Figure 62. The fan plots are computed in vacuum condition in DYMORE. Effect of aerodynamics is incorporatead later using analytical formulas. See sections C.2 and C.2 for more information. In the DYMORE analysis, no aerodynamic loads are applied on the blades. However, centrifugal forces stemming from rotating blade motion are captured. The centrifugal force causes the blade to be stiffer as its angular velocity increases in time. As the blade gets stiffer, its natural frequencies move up. This behavior is evident in Figure 62 for all modes.

Figure 62 and Table 8 show the natural frequencies at various angular speeds. The first flapping frequency at operating angular velocity is close to 1.1P and the lead-lag frequency is close to 0.4P. The computed lead-lag and flapping frequencies

Table 8: The natural frequencies [per rev] at nondimensionalized rotor speeds

Angular Velocity	Flap 1	Flap 2	Flap 3	Flap 4	Lead -lag 1	Lead -lag 2	Feather 1
0.0P	0.135	0.940	2.763	5.533	0.174	4.302	1.815
0.1P	0.173	0.979	2.805	5.582	0.179	4.133	1.873
0.2P	0.264	1.100	2.931	5.716	0.198	4.049	1.924
0.4P	0.454	1.432	3.324	6.155	0.230	4.092	1.985
0.6P	0.657	1.868	3.897	6.830	0.278	4.312	2.060
0.8P	0.854	2.340	4.570	7.664	0.328	4.561	2.158
1.0P	1.066	2.828	5.300	8.604	0.377	4.847	2.280
1.2P	1.270	3.325	6.065	9.615	0.424	5.160	2.420

Figure 63: Application of the disturbance moment

match the design frequencies.

The Auto-Trim feature of the Auto-Trim rotor system uses carefully tailored flexures and a small forward blade sweep to make the rotor capable of self-trimming and virtually immune to gusts. This is accomplished by tailoring the flexures and blades to bring both feathering and flapping natural frequencies close to 1P. As a result of the matched feathering and flapping mode, a situation is created where the feathering motion dynamically reduces the flapping motion. The inter-modal coupling produces a nose down motion as the blade flaps upwards. This system automatically adjusts the cyclic pitch to eliminate any disturbance originating in the blade.

To compute the behavior, the dynamic response of the rotor system to a disturbance loading is simulated. In the DYMORE static analysis mode, a blade pitching moment of 100Nm was applied at the blade tip through the x-axis of the blade as shown in Figure 63.

The deformed configuration of the rotor system as a result of this loading is computed at the rotor speed 1P. The computed deformed configuration is then used

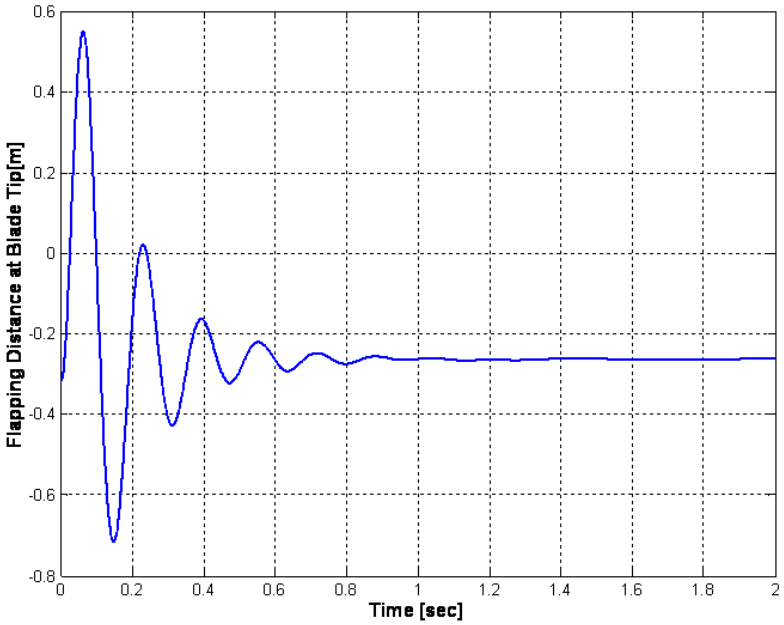


Figure 64: Flapping distance at blade tip

Table 9: Frequencies as result of disturbance [per rev]

Flapping frequency	Feathering frequency	Lead-lag frequency
1.1565	1.1288	0.3778

as the initial condition for a new DYMORE dynamic analysis mode at the time t=0.0. The disturbance loading applied in static analysis is removed at time t=0.0 in order to observe the effect of a temporary gust. A DYMORE dynamic analysis case is run to compute the free vibration response. The aerodynamic loads are also considered for this case. Figure 64 shows the time history the flap-wise displacement at the blade tip. The flap-wise response due to the disturbance is stable.

Figure 65 shows the time histories of the blade tip rotation angles with respect to the hub attached frame. The θ_x, θ_y, and θ_z represent the feather-wise angle, flap-wise angle, and lead-lag-wise angle, respectively. It is possible to calculate the period of response for each motion by measuring the distance between two peaks. The computed frequency terms are shown in Table 9.

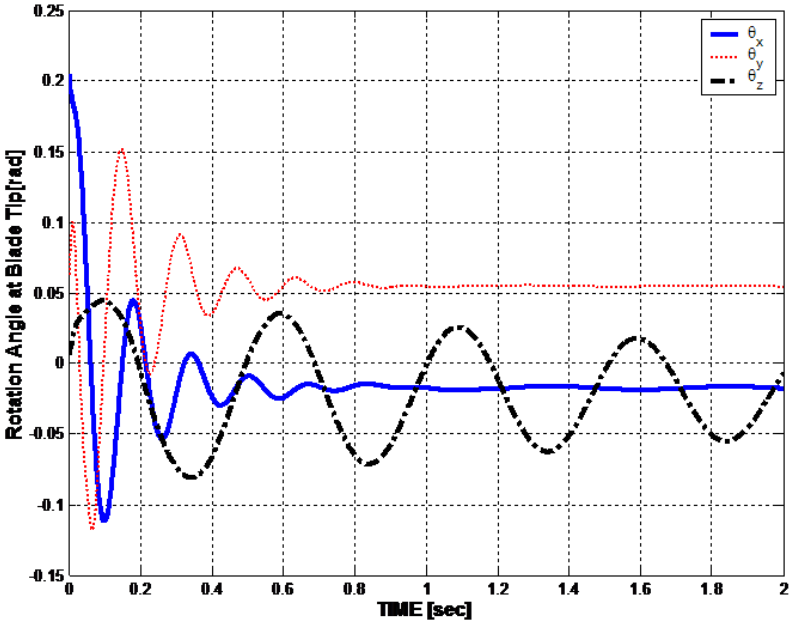


Figure 65: Rotation angles at blade tip

In Table 9, feathering and flapping frequencies are around 1.1P, which is close to 1P. The 1.1P of feathering frequency satisfies the requirement for the auto trim concept.

The blade design is dynamically acceptable. It needs no major modifications in terms of dynamic behavior.

C.2 Validation of Auto Trim

Mathematical modeling of Auto-Trim rotor system is performed in this section. The derivations below were originally conducted by Dr. Jou Young Choi.

Hanson's bearingless Auto-Trim rotor can be classified as a bearingless, therefore also a hingeless rotor. For such a rotor, equations for coupled flapping and feathering motion can be found in numerous references. The below equations are from Bramwell's Helicopter Dynamics textbook [93].

Coupling between torsional and flapping motion is usually investigated due to pitch-flap flutter. This type of flutter is encountered when chordwise center of gravity

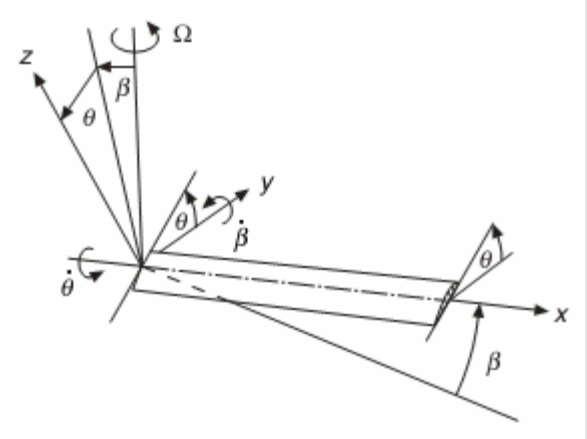


Figure 66: Rotating blade with flapping and torsion

is aft of feathering axis with respect to the leading edge. For most blades, feathering axis is at a quarter-chord away from the leading edge.

Derivation of coupled equations starts with writing equations for blade angular velocity components. Assuming the blade is rigid and hinged at the root with the configuration shown in Figure 66, components of the overall blade angular velocity are

$$\omega_1 = \dot{\theta} + \Omega \sin\beta \approx \dot{\theta} + \Omega\beta \tag{1}$$

$$\omega_2 = -\dot{\beta}\cos\theta + \Omega\sin\theta\cos\beta \approx \dot{\beta} + \Omega\theta \tag{2}$$

$$\omega_3 = \dot{\beta}\sin\theta + \Omega\cos\theta\cos\beta \approx \Omega \tag{3}$$

where θ and β are blade pitching and flapping angles, respectively. Inertia tensor (I) of an object in 3D can be expressed in the form

$$I = \begin{bmatrix} A & F & E \\ F & B & D \\ E & D & C \end{bmatrix}$$

where A, B, C, D and F are mass moments of inertia defined as

$$A = \sum m(y^2 + z^2) \qquad B = \sum m(x^2 + z^2) \qquad C = \sum m(x^2 + y^2)$$

$$D = \sum myz \qquad E = \sum mxz \qquad F = \sum mxy$$

m being mass of a particle in the object. Angular momentum components of a body are given in Ref. [93] as

$$h_1 = A\omega_1 - F\omega_2 - E\omega_3 \tag{4}$$

$$h_2 = B\omega_2 - D\omega_3 - F\omega_1 \tag{5}$$

$$h_2 = C\omega_3 - E\omega_1 - D\omega_2 \tag{6}$$

A helicopter blade can be assumed to be a slender body; that is, its thickness is much less than its chord length or span. Hence the terms D and E may be negligible compared to terms containing dimensions on x and y axes. Applying this simplification, angular momentum components become

$$h_1 = A\omega_1 - F\omega_2 \tag{7}$$

$$h_2 = B\omega_2 - F\omega_1 \tag{8}$$

$$h_2 = C\omega_3 \tag{9}$$

Differentiating with respect to time and retaining only first order terms,

$$\dot{h_1} = A\ddot{\theta} + A\Omega^2\theta + F\ddot{\beta} + F\Omega^2\beta$$

$$\dot{h_2} = -B\ddot{\beta} - B\Omega^2\theta - F\ddot{\theta} - F\Omega^2\theta$$

Coupled equations of motion with elastic stiffnesses about rotation axes are,

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\psi^2} + \nu_1^2\theta + \frac{F}{A}\left(\frac{\mathrm{d}^2\beta}{\mathrm{d}\psi^2} + \beta\right) = \frac{L_A}{A\Omega^2} \tag{10}$$

$$\frac{\mathrm{d}^2\beta}{\mathrm{d}\psi^2} + \lambda_1^2\beta + \frac{F}{B}\left(\frac{\mathrm{d}^2\theta}{\mathrm{d}\psi^2} + \theta\right) = \frac{M_A}{B\Omega^2} \tag{11}$$

where L_A and M_A are aerodynamic torsional and flapping moments, respectively, and $\nu_1\Omega$ and $\lambda_1\Omega$ are uncoupled torsional and flapping frequencies in vacuum. Equations 10 and 11 need to be solved in order to show the effects of Auto-Trim behavior on the fan plots.

The uncoupled feathering frequency in air can found by rewriting Equation 11 as:

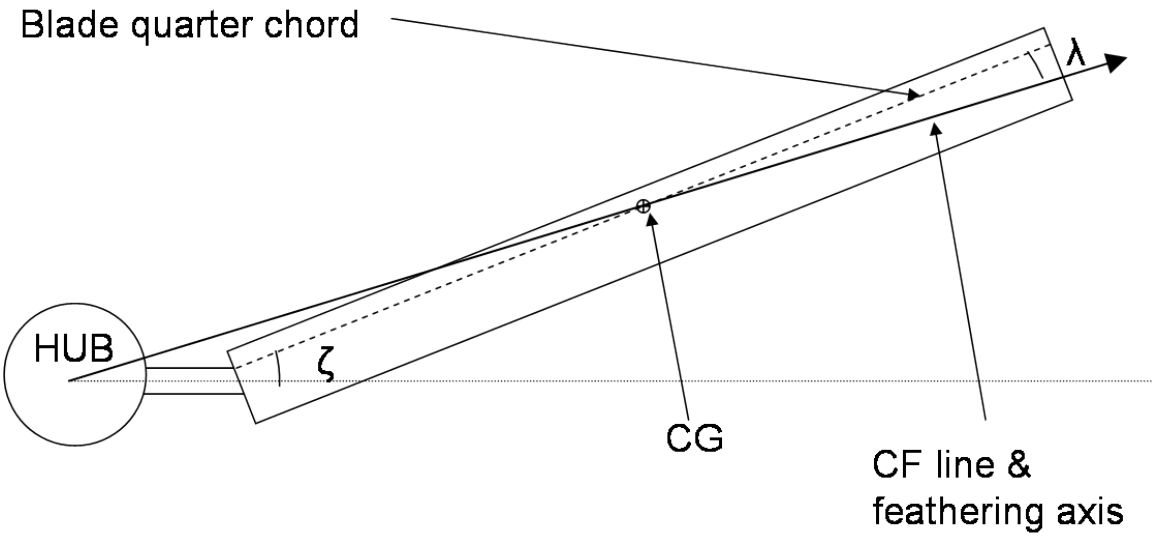


Figure 67: Exaggerated configuration of Auto-Trim rotor blade, looking from top

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\psi^2} + \nu_1^2\theta = \frac{L_A}{A\Omega^2} \tag{12}$$

Let ω_0 be feathering frequency at zero rpm. From Bramwell's Helicopter Dynamics [93],

$$\nu_1^2 = \frac{\omega_0^2}{\Omega^2} + \frac{\Omega^2}{\Omega^2} = 1 + \frac{\omega_0^2}{\Omega^2} \tag{13}$$

Equation 12 then becomes

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\psi^2} + \left(1 + \frac{\omega_0^2}{\Omega^2}\right)\theta = \frac{L_A}{A\Omega^2} \tag{14}$$

Equation 14 can be solved once L_A term is determined. This term can be calculated from moment balance about blade span axis. Figure 67 shows exaggerated blade configuration and necessary angles. CF line is assumed to be aligned with feathering axis.

Location of CG point along the span is an important parameter for calculations. The exact location depends on blade structure and tip mass. The CG location measured from blade root is expressed in terms of blade length, αR_{blade}, where α is a constant between zero and one. Figure 68 shows that and other major dimensions. Letting a be aerodynamic lift constant, ρ be air density, and c be chord length, equation for calculating the torsional moment L_A can be expressed as:

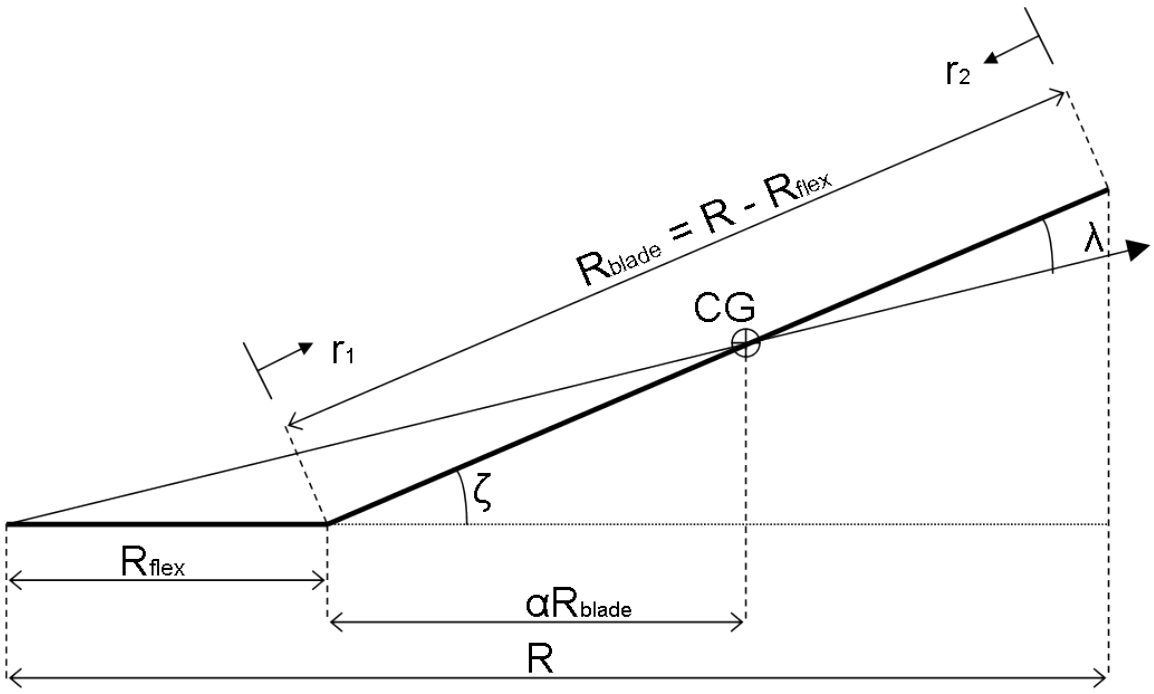


Figure 68: Simplified Auto-Trim rotor blade

$$\frac{L_A}{\theta} = -\int_0^{\alpha R_{blade}} \left(\frac{1}{2}\rho ac\Omega^2\right) \vec{r_1}^2 r\sin\lambda \,\mathrm{d}r + \int_0^{1-\alpha R_{blade}} \left(\frac{1}{2}\rho ac\Omega^2\right) \vec{r_2}^2 r\sin\lambda \,\mathrm{d}r \quad (15)$$

where coordinates of $\vec{r_1}$ and $\vec{r_2}$ in 2D lead-lag plane are defined as

$$\vec{r_1} = (R_{flex} + (\alpha R_{blade} - r)\cos\zeta,\ (\alpha R_{blade} - r)\sin\zeta) \quad (16)$$

$$\approx (R_{flex} + (\alpha R_{blade} + r), (\alpha R_{blade} + r)\zeta) \quad (17)$$

$$\vec{r_2} \approx (R_{flex} + (\alpha R_{blade} + r), (\alpha R_{blade} + r)\zeta) \quad (18)$$

Their squares will therefore be

$$\vec{r_1}^2 \cong (R_{flex} + \alpha R_{blade} - r)^2 + (\alpha R_{blade} - r)^2\zeta^2 \quad (19)$$

$$\vec{r_2}^2 \cong (R_{flex} + \alpha R_{blade} + r)^2 + (\alpha R_{blade} + r)^2\zeta^2 \quad (20)$$

Substituting these values in Equation 15 and rearranging,

$$\begin{aligned}\frac{L_A}{\theta} = &\left(\frac{1}{2}\rho ac\Omega^2\lambda\right) \\ &[-\int_0^{\alpha R_{blade}} \left[(R_{flex} + \alpha R_{blade} - r)^2 + (\alpha R_{blade} - r)^2\zeta^2\right] r\,\mathrm{d}r \\ &+ \int_0^{1-\alpha R_{blade}} \left[(R_{flex} + \alpha R_{blade} + r)^2 + (\alpha R_{blade} + r)^2\zeta^2\right] r\,\mathrm{d}r]\end{aligned} \quad (21)$$

The integration is carried out using MATLAB. Let

$$Q = -\int_0^{\alpha R_{blade}} \left[(R_{flex} + \alpha R_{blade} - r)^2 + (\alpha R_{blade} - r)^2 \zeta^2\right] r \, \mathrm{d}r + \int_0^{1-\alpha R_{blade}} \left[(R_{flex} + \alpha R_{blade} + r)^2 + (\alpha R_{blade} + r)^2 \zeta^2\right] r \, \mathrm{d}r \tag{22}$$

From MATLAB,

$$Q = -\frac{1}{12} R_{blade}^2 (12 R_{flex} \alpha R_{blade} + 12 R_{flex}^2 \alpha + 4\zeta^2 \alpha R_{blade}^2 - 3R_{blade}^2 + 4\alpha R_{blade}^2 - 8 R_{flex} R_{blade} - 6 R_{flex}^2 - 3\zeta^2 R_{blade}^2) \tag{23}$$

Substituting Q into Equation 21 and adding inertial terms,

$$\frac{L_A}{A\Omega^2} = \frac{\theta}{A\Omega^2} Q \left(\frac{1}{2}\rho a c \Omega^2 \lambda\right) = \frac{Q\theta}{A}\left(\frac{1}{2}\rho a c \lambda\right) \tag{24}$$

Substituting Equation 24 into Equation 14

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}\psi^2} + \left[1 + \frac{\omega_0^2}{\Omega^2} - \frac{Q}{A}\left(\frac{1}{2}\rho a c \lambda\right)\right] = 0 \tag{25}$$

Finally, the feathering frequency in air becomes

$$\frac{\omega^2}{\Omega^2} = 1 + \frac{\omega_0^2}{\Omega^2} - \frac{Q}{A}\left(\frac{1}{2}\rho a c \lambda\right) \tag{26}$$

Change in feathering frequency can be plotted and compared to 1P line throughout the angular velocity range of the rotor. Such a plot is depicted in Figure 69. The frequency change behavior depends on spanwise location of the CG (value of α) and forward sweep angle (ζ). Figures 70 and 71 show the change in first flapping frequency.

C.3 Hanson's Model

Tom Hanson has a different approach for modeling his rotor design mathematically. According to his book [69], the relation between flapping and feathering motion can be modeled similar to a dynamic vibration absorber with two masses connected to each other with a spring, as shown in Figure 72 [69]. In the figure, mass M_1 remains

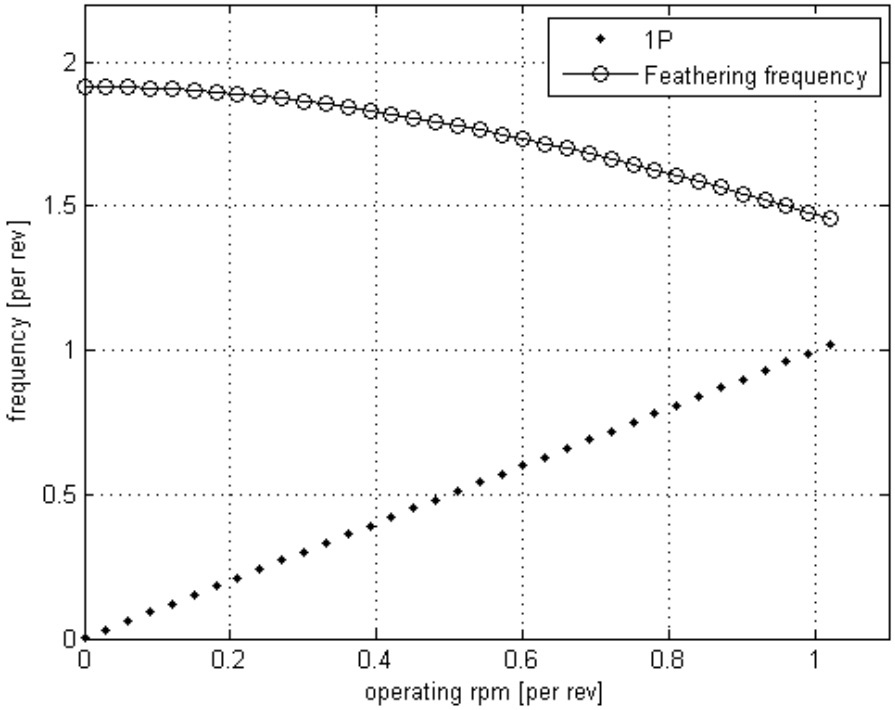


Figure 69: Feathering frequency at $\alpha = 0.58$ and $\zeta = 2$ degrees

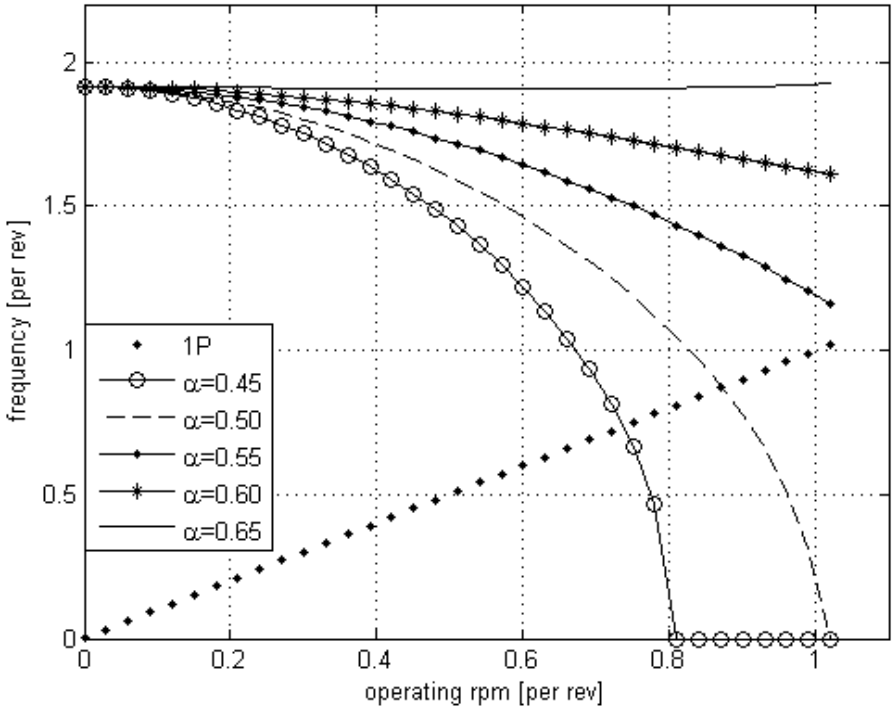


Figure 70: Feathering frequencies at $\zeta = 2$ degrees

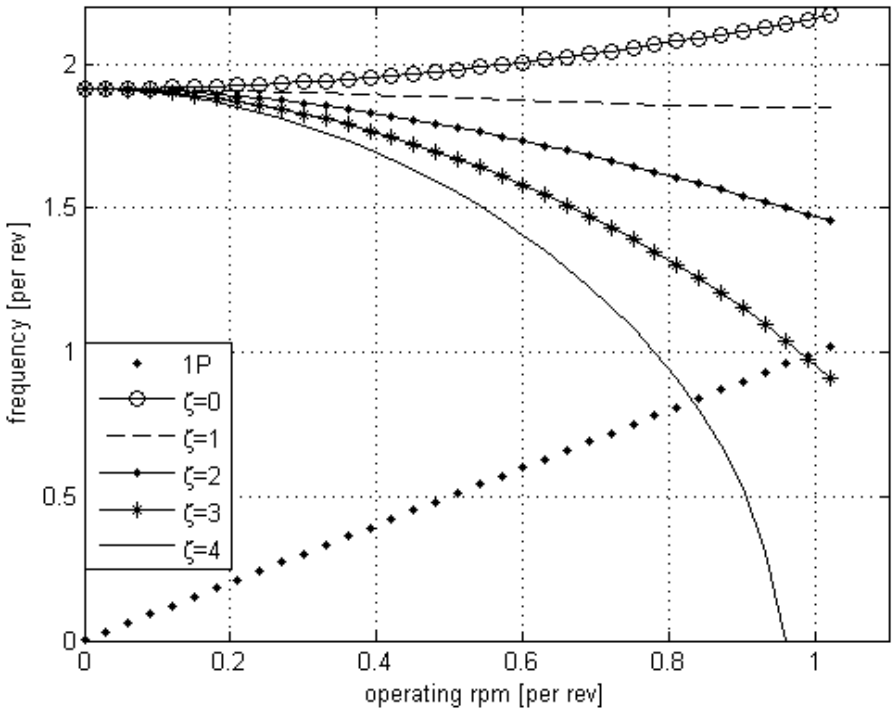


Figure 71: Feathering frequencies at $\alpha = 0.58$

still under the effect of forcing function $P_0 \sin \omega t$. This is because the mass M_2 is excited by another forcing function, $F \sin \omega t$, and the varying forces applied by the spring under M_1 cancel out $P_0 \sin \omega t$. In order for this system to work, the forcing functions must be equal but opposite in amplitude, i.e., $P_0 = F$.

The Auto-Trim system is modeled in a similar way. For instance when there is an upwards aerodynamic disturbing force present, blade is pitched downwards to prevent flapping. The blade mass is analogous to M_1 in the vibration absorber system. M_2 is analogous to the feathering inertia of the blade. The exciting force replacing $P_0 \sin \omega t$ is a 1P excitation. Spring stiffnesses K_1 and K_2 are now defined as flapping and feathering stiffnesses, K_β and K_θ, respectively. A schematic of the system is shown in Figure 73.

Since the flapping motion is created by a 1P excitation force, the feathering natural frequency must be 1P as well in order for Auto-Trim to work. The feathering frequency can be obtained by using the formula [69];

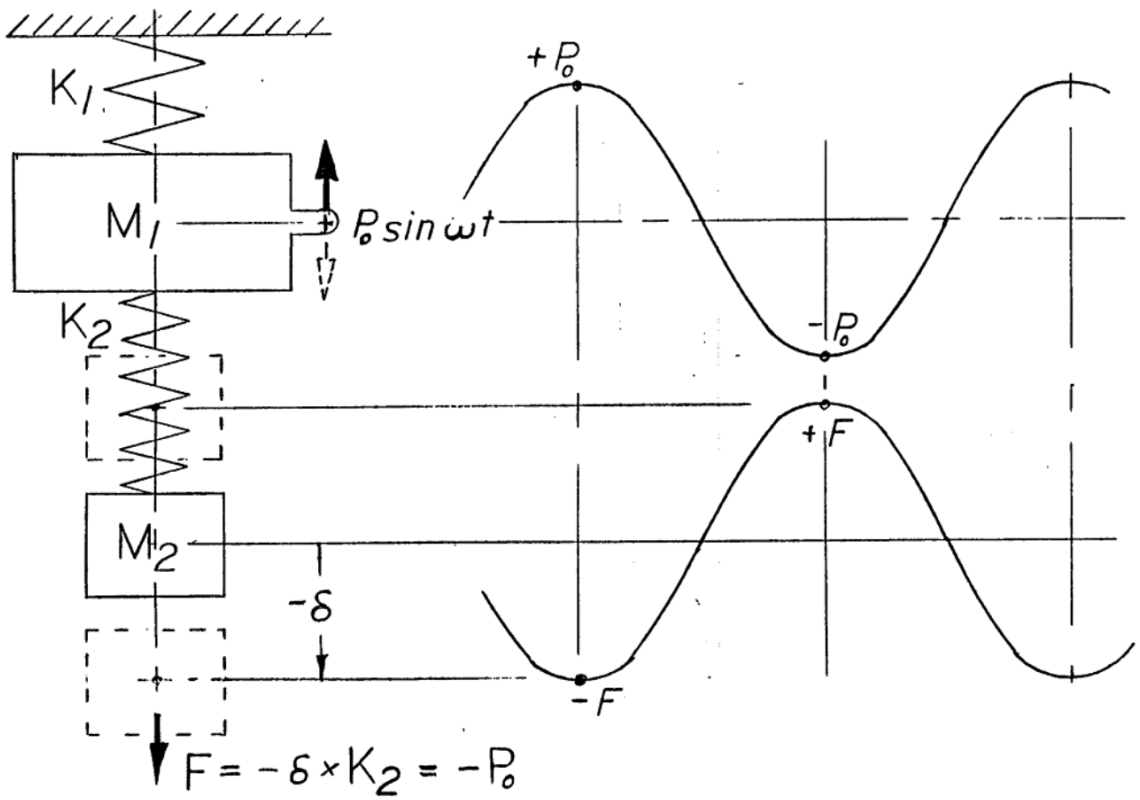


Figure 72: A dynamic vibration absorber

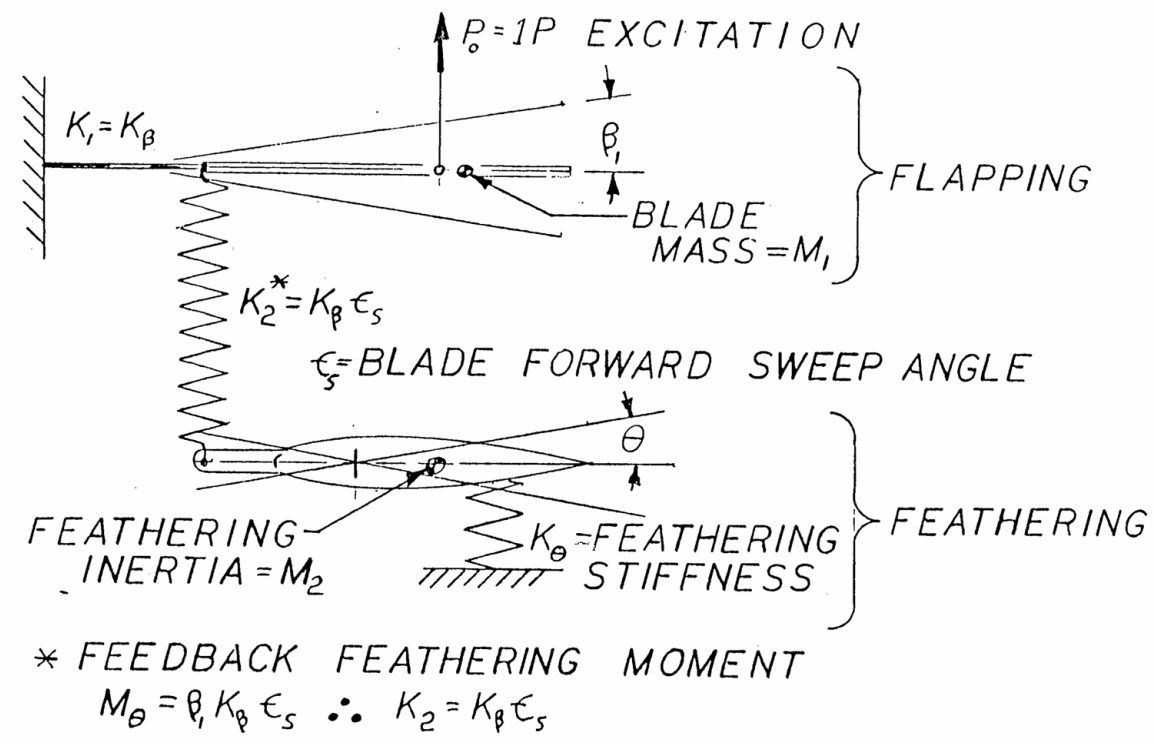


Figure 73: A vibration-absorber-like model of the Auto-Trim system.

Table 10: Information gathered from VABS and DYMORE

Parameter	value	units
I_h	0.01523	slug ft^2
I_v	0.000247	slug ft^2
$k_{torsional}$ at $\Omega = 0$ rad/s	252.649	lb ft /rad
$k_{torsional}$ at $\Omega = 33.33$ rad/s	352.739	lb ft /rad

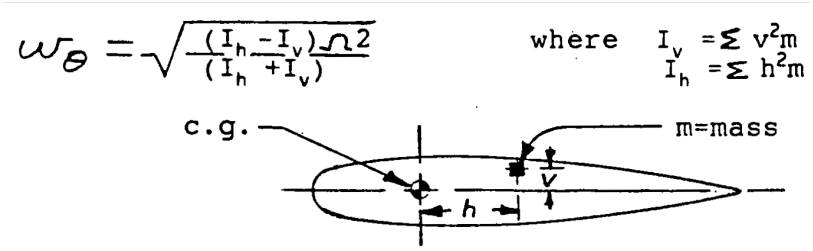


Figure 74: Hanson's formula for calculating base feathering frequency

$$\omega_n = \sqrt{\frac{k_b}{I_{polar}}} = \sqrt{\frac{k_{base} + k_{torsional} + k_{spring}}{I_h + I_v}} \tag{27}$$

where;

w_0: torsional natural frequency of the blade

k_b: blade stiffness

I_p: polar mass moment of inertia

I_v, I_h: vertical and horizontal mass moments of inertia, respectively

k_{base}: base stiffness

$k_{torsional}$: elastic torsional stiffness

k_{spring}: negative spring stiffness produced by blade forward sweep

The information presented in Table 10 is obtained from VABS and DYMORE. Using the last two rows of the table, torsional stiffness at any given angular speed may be approximated with a linear function: $k_{torsional} = 3.0\ \Omega + 252.65$. The rotor gets stiffer because of CF effect. The base stiffness and negative spring stiffness at operating rpm can be found by using Figure 74 and the formulas [69];

$$k_{base} = (I_h - I_v)\Omega^2$$
$$= 0.049 \cdot 33.33^2 = 54.61 \text{ lb ft}^2$$

$$\begin{aligned} k_{spring} &= \sum \left[\frac{\text{Lift}}{\theta}\delta_s + \frac{\text{Lift}}{\theta}\delta_x \right] \\ &= \frac{cRV_{tip}^2}{1472}\left[-3.30\delta_s + \sum \left(\frac{R_x}{R} \right)^2 \delta_x \right] \\ &= 4034.6 \left[-3.30\delta_s + \sum \left(\frac{R_x}{R} \right)^2 \delta_x \right] \\ &= 4034.6\left[-3.30\delta_s - 1.392\delta_s \right] \qquad \text{(for CG located at 0.58R)} \\ &= 4034.6 \cdot (-0.06647) \qquad (\text{for } \delta_s = 0.17 \text{ in}) \\ &= -295.55 \text{ ftlb/rad} \end{aligned} \tag{28}$$

where δ is the distance from quarter chord to CF vector. δ_s is that distance at the blade-flexure attachment point. It is assumed to be 0.17 inches. R_x is a spanwise distance on the blade measured from hub center. The blade is sliced into equal lengths of $0.1R$ for ease of calculations. Moments acting on each slice are summed in the above equation for k_{spring}. Substituting all those values into Equation 27, ω_n is found to be closer to 1P at operating rpm in Equation 29. Relation between Ω and ω_n is depicted in Figure 75. Effects of spanwise CG location and blade forward sweep angle on feathering frequency are shown in Figure 76 and Figure 77, respectively.

$$\begin{aligned} \omega_n &= \sqrt{\frac{54.61 + 352.739 - 295.55}{0.0499 + 0.00081}} \\ &= 52.37 \text{ rad/s } = 1.57 \text{ P} \end{aligned} \tag{29}$$

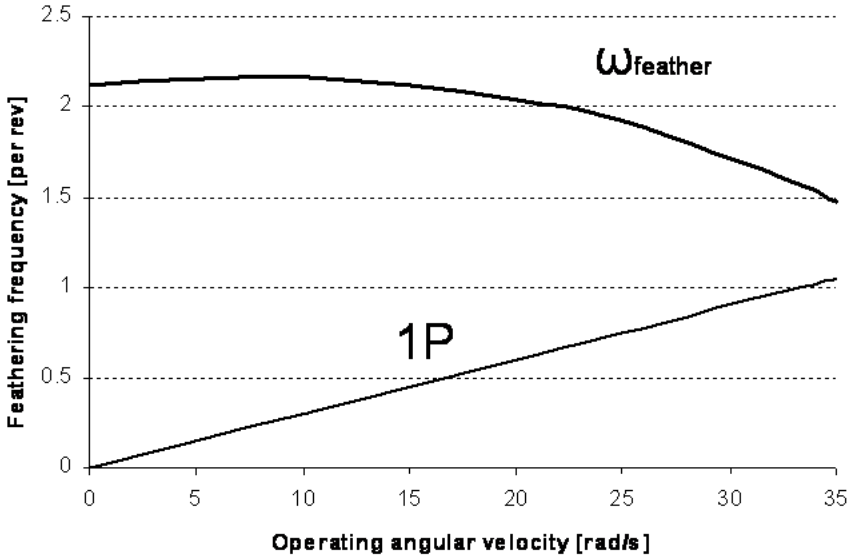


Figure 75: Effect of rpm on feathering frequency ($\alpha = 0.58$, $\zeta = 2°$)

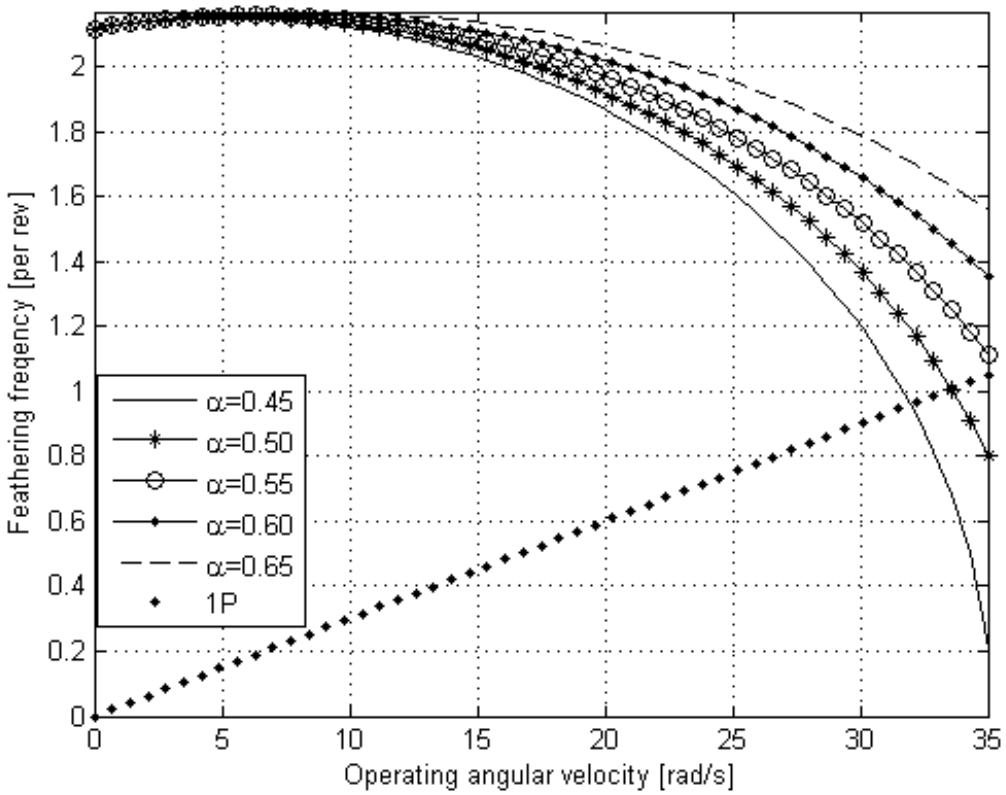


Figure 76: Effect of spanwise blade CG location on feathering frequency ($\zeta = 2°$)

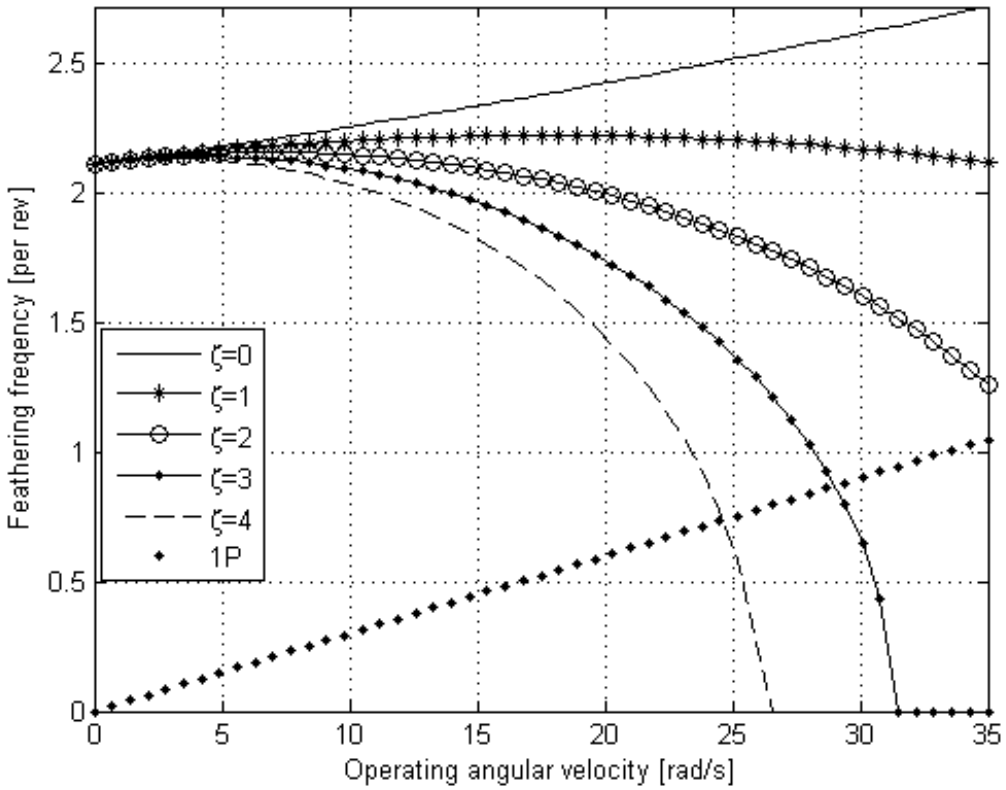


Figure 77: Effect of forward sweep angle on feathering frequency ($\alpha = 0.58$)

APPENDIX D

FLAP FATIGUE CALCULATIONS

D.1 Flapping Moment for a Simple Beam Crossection

Effect of a Moment M applied to a beam section about the z-axis as shown in Figure 78 is given in ref [95] as,

$$M = -\int y\sigma_x \, \mathrm{d}A \tag{30}$$

where $\mathrm{d}A$ is the crossectional area of the beam, y is the vertical coordinate of an infinitesimal particle, and σ_x is stress on the crossection at x. The negative sign is a result of sign convention. By Hooke's law,

$$\sigma_x = \mathrm{E}\epsilon_x \tag{31}$$

where ϵ_x is the strain in x-direction. For a beam under a pure bending moment M, ϵ_x may be assumed to simplify to:

$$\epsilon_x = -y\frac{\mathrm{d}\Phi_z}{\mathrm{d}x} = -y\frac{\mathrm{d}^2 u_y}{\mathrm{d}x^2} \tag{32}$$

The negative sign is again a result of sign convention. Substituting this into Equation 30, the moment M becomes

$$M = \int y^2 \mathrm{E}\frac{\mathrm{d}^2 u_y}{\mathrm{d}x^2} \, \mathrm{d}A \tag{33}$$

where E is Young's Modulus. If E is assumed to be constant for the entire crossection, it can be moved out of the integral. Likewise, the curvature, $\mathrm{d}^2 u_y / \mathrm{d}x^2$, is also assumed constant. Equation 33 now becomes

$$M = \mathrm{E}\frac{\mathrm{d}^2 u_y}{\mathrm{d}x^2} \int y^2 \, \mathrm{d}A \tag{34}$$

Moment of inertia of the beam about x-axis (I_x) can be written as

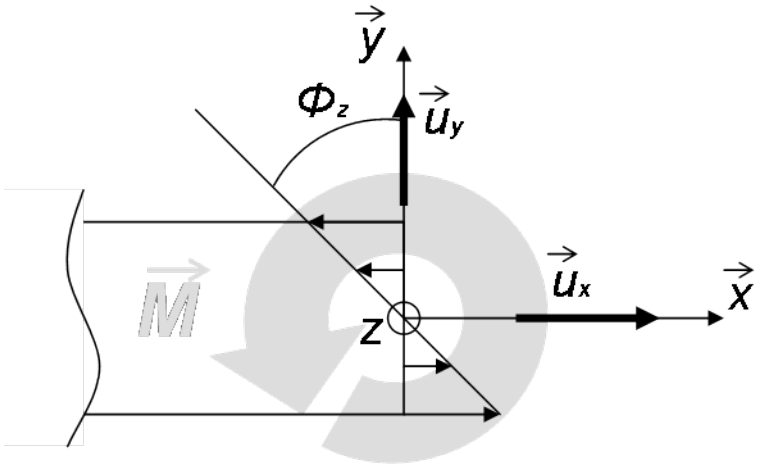


Figure 78: Moment M being applied to a beam section

$$I_x = \int y^2 \, \mathrm{d}A \tag{35}$$

Therefore,

$$M = \mathrm{E}I_x \frac{\mathrm{d}^2 u_y}{\mathrm{d}x^2} \qquad \frac{M}{\mathrm{E}I_x} = \frac{\mathrm{d}^2 u_y}{\mathrm{d}x^2}$$

Using Equation 32,

$$\frac{M}{\mathrm{E}I_x} = \frac{y \, \mathrm{d}^2 u_y}{y \, \mathrm{d}x^2} = \frac{\epsilon_x}{y}$$

The maximum strain is expected to occur on the upper or lower surface of the flexure for the given loading condition. Thus,

$$\frac{M}{\mathrm{E}I_x} = \frac{\epsilon_{xmax}}{y_{max}} \tag{36}$$

D.2 Auto-Trim Rotor Flexure

The flexure geometry in the case of Auto-Trim rotor is more complicated than the beam section considered in the previous section. The sectional geometry is depicted in Figure 79. The Flexure is made of a composite material: generic S-Glass/Epoxy unidirectional prepreg. Properties of this material are listed in Table 7. Since it is unidirectional (i.e. layup angle of 0°), it is assumed to have uniform E over the entire crossection for loading conditions along the x-axis.

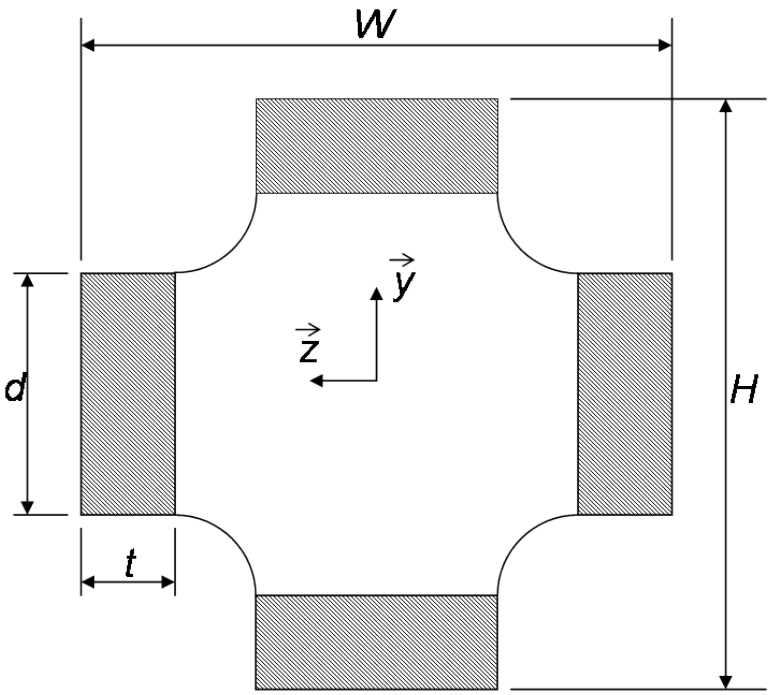


Figure 79: Crossection of the flexure

S-Glass/epoxy is used in the shaded regions of Figure 79. The core of the structure is filled with syntactic epoxy foam (SEF). The effects of this filling material to the flexure structural properties are ignored.

The crossection is symmetric about y and z-axes. Moment of inertia about x-axis for this crossection is

$$I_x = \int y^2 \, \mathrm{d}A \tag{37}$$

$$= 2\left[\frac{td^3}{12} + \frac{dt^3}{12} + \frac{2dt}{2}\left(\frac{H}{2} - \frac{t}{2}\right)^2\right] \tag{38}$$

$$= \frac{td}{2}\left[\frac{d^2 + t^2}{3} + (H - t)^2\right] \tag{39}$$

The maximum allowable flapping moment the rotor can withstand without risking fatigue damage can be calculated. According to ref. [94], for S-glass/epoxy, endurance limit for tension-tension load cycles is about σ_{allow}=0.21 GPa (N$\approx 10^7$). With E=43.0 GPa from Table 7, maximum allowable strain (ϵ_{allow}) is;

Table 11: Dimensions at the flexure root crossection as shown in Figure 79

W	H	d	t	y_{max}	units
2.89	3.37	0.60	0.20	1.44	in

$$\begin{aligned}\sigma_{allow} &= \mathrm{E}\ \epsilon_{allow} \\ \epsilon_{allow} &= \frac{\sigma_{allow}}{\mathrm{E}} \\ &= \frac{0.21}{43} \\ &= 0.0048 = 0.48\ \%\end{aligned}$$

Substituting this value with ϵ_{xmax} in Equation 36,

$$\frac{M_{allow}}{\mathrm{E}I_x} = \frac{\epsilon_{allow}}{y_{max}} \tag{40}$$

A simplified sketch of one blade-flexure pair is shown in Figure 80. The applied bending moment would be most intense at the root of the flexure; point A. At that point, the dimensions of flexure crossection are given in ref. [69] as shown in Table 11

With those dimensions substituted into equations 39 and 40, and taking E= 46 10^9Pa from Table 7,

$$\begin{aligned}I_x &= 0.61\ \mathrm{in}^4 = 2.54\ 10^{-}7\ \mathrm{m}^4 \\ \frac{M_{allow}}{11697.33} &= \frac{0.0048}{0.036} \\ M_{allow} &= 1561.85\ \mathrm{Nm} = 1151.81\ \mathrm{lbf\ ft}\end{aligned}$$

In order to find the maximum allowable flapping angle β_{allow}, the M_{allow} is created at point A. An equivalent force of 397.8 lbf is applied at point B in Figure 80, which is 2.89 ft away from point A. A static case of DYMORE simulation is run and deflection angle at point B is measured. The deflection is found to be 9.42°. Noting that there was a precone angle of $\beta_0 = 3.5°$, the maximum allowable flapping angle is found to be:

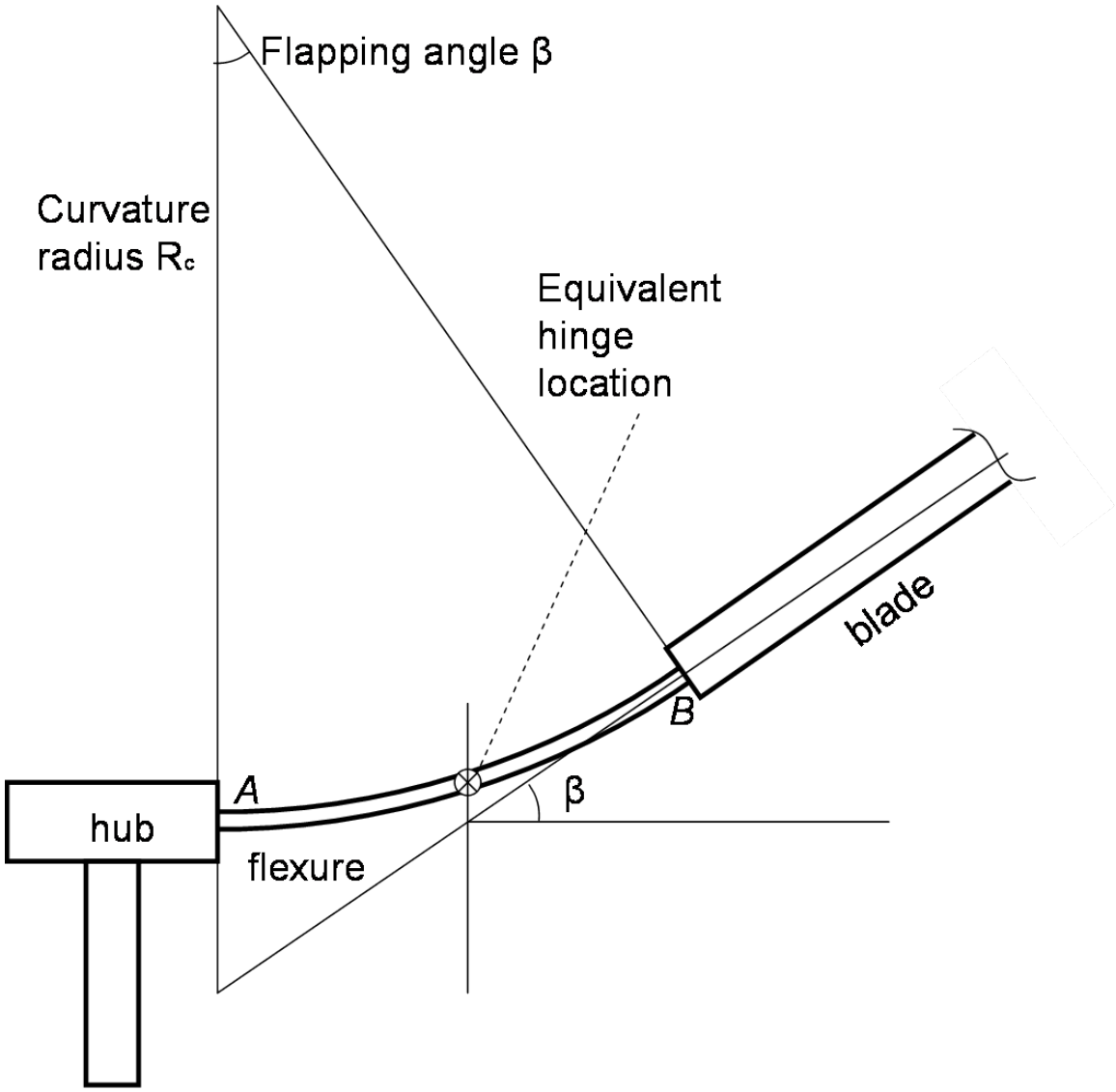


Figure 80: Simplified flexure geometry in bending

$$\beta_{allow} = \pm(9.42 - 3.5) = \pm 5.59^{\circ}$$

Having the information of β_{allow}, it is possible to find the gust speed that would produce such flapping. Under normal operating conditions in hover, CF on the blade exerts a force to reduce flapping angle. This force enables higher aerodynamic lift forces to be generated on the blade without bending it beyond the allowable angle. Setting $\beta = \beta_{allow}$ as constant, maximum aerodynamic moment the rotor can withstand within fatigue limits can be found. The total flapping moment at point A as given in Figure 81 is,

$$\vec{M_{total}} = \vec{M_A} - \vec{CF} \times \vec{d_{CF}} \tag{41}$$

where $\vec{CF}$ is the centrifugal force applied to the blade, and $\vec{M_A}$ is the aerodynamic flapping moment generated on the blade. $\vec{d_{CF}}$ is the moment arm of the centrifugal force. Their values are calculated as shown below. The assumptions and required parameter values are given in Table 12. Using the information in Table 12,

$$\begin{aligned} CF &= m_{blade}\Omega^2 \left[\left(\frac{R_{blade}}{2} + \frac{R_{flex}}{2} \right) \cos\beta + \frac{R_{flex}}{2} \right] \\ &= 26.7 \cdot 33.33^2 \cdot 10.19 = 302350 \text{ lbs} \end{aligned}$$

$$\begin{aligned} M_A = \int_0^{R_{blade}} & \frac{1}{2}\rho c c_l \left[\Omega((r + \frac{R_{flex}}{2})\cos\beta + \frac{R_{flex}}{2}) + V_{gust} \right]^2 \\ & \cdot \left(r + \frac{R_{flex}}{2} + \sin\beta \frac{R_{flex}}{2} \right) \mathrm{d}r \\ = \int_0^{14.83} & \frac{1}{2} 0.0745 \cdot 0.9167 \cdot 2 \cdot \pi \cdot \frac{\pi}{180}(-12\frac{r}{14.83} + 11) \\ & \cdot \left[33.33 \left((r + 0.45)0.9865 + 0.45 \right) + V_{gust} \right]^2 (r + 1.45 + 0.163 \cdot 1.45) \, \mathrm{d}r \\ = 880.88 &+ 778.56 \, V_{gust} + 2.03(95.85 + \, V_{gust})^2 \text{ ft lb} \end{aligned}$$

$$\begin{aligned} d_{CF} &= \sin\beta \cdot \left(\frac{R_{blade}}{2} + \frac{R_{flex}}{2} \right) \\ &= 0.163 \cdot (7.42 + 1.45) = 1.45 \text{ ft} \end{aligned}$$

Table 12: Assumptions and necessary parameter values

Blade mass (m_{blade})	0.15 lb/in $\cdot$ R_{blade}	26.7 lbs
Equivalent hinge offset location from flexure root	$R_{flex}/2$	1.45 ft
Blade CG location from blade root	$R_{blade}/2$	7.42 ft
Application point of CF	Blade CG	
Blade structural deformation	Insignificant compared to flexure deformation	
Blade twist angle variation	$\theta_{tw} = -12\ r/R_{blade} + 11$ degrees	
Air density (ρ)	0.0745 lb/ft^3	
Lift coefficient (c_l)	$2\pi\ \theta_{tw}$	

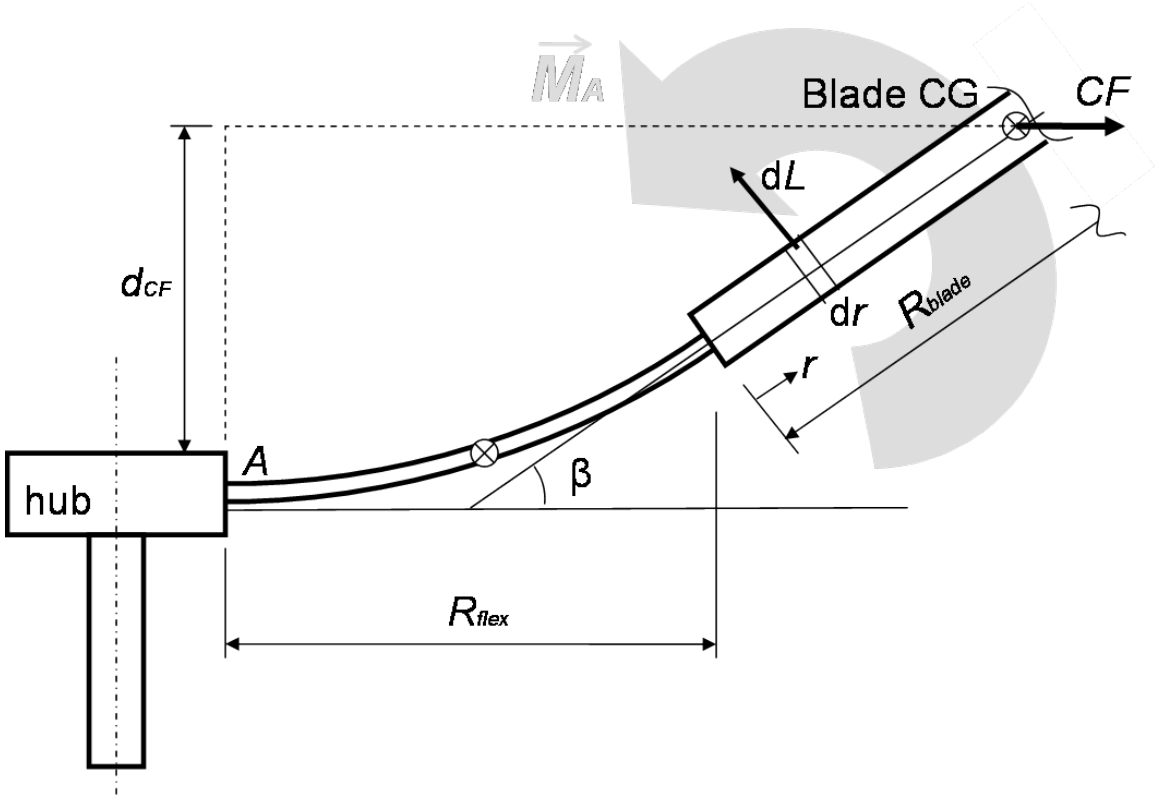


Figure 81: Application of moments during operation in hover

Using the values calculated in the above equations, the gust speed V_{gust} can be found by equating total moment M_{total} to the allowable moment M_{allow} and solving for V_{gust}. Using MATLAB;

$$M_{total} = M_A - CF \cdot d_{CF}$$

$$M_{allow} = M_{total} = \left[880.88 + 778.56\ V_{gust} + 2.03(95.85 + \ V_{gust})^2\right] - 438710$$

$$1151.81 = \left[880.88 + 778.56\ V_{gust} + 2.03(95.85 + \ V_{gust})^2\right] - 302350$$

$$\Rightarrow \quad V_{gust} \cong 165.6 \text{ ft/s} \cong 112.9 \text{ mph} \cong 98 \text{ kts}$$

Since the rotor is not supposed to hover under conditions in which wind speeds may find 113 mph, it can be safely assumed that the flexure is within endurance limits for hover. For forward flight speeds at about a hundred knots, however, variations in headwinds might be an issue for fatigue considerations.

APPENDIX E

COMPUTER CODES

E.1 MATLAB Code Used in Modelcenter for Extracting DYMORE Data

```
% switch to the directory where the m-file
% is stored. If the m-file is in your path,
% you may delete this line
 cd F:\ModelCenterFolder\KU_haiying\DYMORE\LHYdym\
 close all
% run the m-file. If your function
% requires arguments, you should manually
% enter them here.
% variable: leadlag1 double output
% variable: flap1 double output
% variable: flap2 double output
% variable: feather double output
% variable: static_stability double output
% variable: dymore_worked double input
% variable: dymore_worked_dyn double input
% variable: omg_rad_sec double input
% variable: Lblade double input
% variable: L_flex_2B_ft double input
% variable: Mtip double input
% variable: air_density double input
% variable: chord_ft double input
% variable: epsilon double input
% variable: lift_curve_slope double input
% variable: flap_disp array input
% variable: omg_rad_sec double input
% array: freq1 double output
ft2m=0.3048;
%EIGENVALUES_____________________________________________________________
clear step eigen_val eigenval ei y x fr freq leadlag1 flap1 flap2 leadlag2
flap3 feather flap4 leadlag3
aa rho omega aw bw at
eigen_val=[];
n=1;
fid = fopen('LHY_static.eig', 'rt');
while  feof(fid)==0
    tline = fgetl(fid);
    matches1 = findstr(tline, '******:');
    if matches1 > 0
        step=sscanf(tline, '%*s %2f' );
    end
    matches2 = findstr(tline, 'EigenValue:');
    if matches2 > 0
        ei=sscanf(tline, '%*s %*f %24f' );
        eigen_val(n,1)=step(1);
        eigen_val(n,2) = ei(1);
        eigenval(step(1),mod((n-1),4)+1) = ei(1);
        n=n+1;
    end
end
fclose(fid);
siz=size(eigenval);
```

```
P=linspace(0,1,siz(1))
time_s=linspace(0,siz(1)/10,siz(1))
eigenval=eigenval/omg_rad_sec;
figure(1)
plot(time_s,eigenval(:,1),'k','linewidth',2)
hold
plot(time_s,eigenval(:,2),'.k','linewidth',2)
plot(time_s,eigenval(:,3),'.-k','linewidth',2)
plot(time_s,eigenval(:,4),'--k','linewidth',2)
plot(time_s,P,'--k')
plot(time_s,2*P,'k')
plot(time_s,3*P,'--k')
plot(time_s,4*P,'k')
plot(time_s,5*P,'--k')
legend('1st Lead-Lag','1st Flap','1st Feather','2nd Flap','1P','2P','3P',
'4P','5P','Location','NorthWest')
xlabel('Time (s)')
ylabel('Frequency (per rev)')
grid on
tim=round(step(1)/6.2*6);
freq1=[];
for n=1:length(eigen_val)
    if eigen_val(n,1)==tim
        freq1=[freq1 eigen_val(n,2)];
    end
end
freq=freq1/omg_rad_sec;
leadlag1=freq(1);
flap1=freq(2);
feather=freq(3);
flap2=freq(4);

if flap1<=1 | leadlag1<=0.3
static_stability=1000
else
static_stability=(flap1-1)+(1/abs(0.55-leadlag1))+2*(1/abs(round(flap2)-flap2))
end

save('OpenThisInExcel.txt', 'eigenval', '-ASCII')

%_____________________________________________________________________


dymores=dymore_worked+dymore_worked_dyn;
if dymores>=5.9

%    variable: dynamic_stability double output
%    variable: feather_period double output
%    variable: flap_period double output
%    array: time double input
%    array: LL_disp double input
%    array: flap_disp double input
%    array: tw_disp double input

clear dynamic_stability difp store bb_dis_3 bb_dis_2 bb_dis_4 aa time_s

bb_dis_2=LL_disp;
bb_dis_3=flap_disp;
bb_dis_4=tw_disp;

%finding peaks
 peaks=[];
 ptime=[];
 peaks2=[];
 ptime2=[];
 peaks4=[];
 ptime4=[];

six1=find(time<=7.0 & time>=6.5);
six=six1(1);
for tt=six:0.99*length(bb_dis_3)
```

```
    if bb_dis_3(tt)>bb_dis_3(tt-1) & bb_dis_3(tt)>bb_dis_3(tt+1)
        peaks=[peaks bb_dis_3(tt)];
        ptime=[ptime time(tt)];
    end
    if bb_dis_2(tt)>bb_dis_2(tt-1) & bb_dis_2(tt)>bb_dis_2(tt+1)
        peaks2=[peaks2 bb_dis_2(tt)];
        ptime2=[ptime2 time(tt)];
    end
    if bb_dis_4(tt)>bb_dis_4(tt-1) & bb_dis_4(tt)>bb_dis_4(tt+1)
        peaks4=[peaks4 bb_dis_4(tt)];
        ptime4=[ptime4 time(tt)];
    end
end

lows=[];
ltime=[];
lows2=[];
ltime2=[];
lows4=[];
ltime4=[];
for tt=six:0.99*length(bb_dis_3)
    if bb_dis_3(tt)<bb_dis_3(tt-1) & bb_dis_3(tt)<bb_dis_3(tt+1)
        lows=[lows bb_dis_3(tt)];
        ltime=[ltime time(tt)];
    end
    if bb_dis_2(tt)<bb_dis_2(tt-1) & bb_dis_2(tt)<bb_dis_2(tt+1)
        lows2=[lows2 bb_dis_2(tt)];
        ltime2=[ltime2 time(tt)];
    end
    if bb_dis_4(tt)<bb_dis_4(tt-1) & bb_dis_4(tt)<bb_dis_4(tt+1)
        lows4=[lows4 bb_dis_4(tt)];
        ltime4=[ltime4 time(tt)];
    end
end


fp=[peaks; ptime];
fl=[lows; ltime];
f=[fp fl];
f=f';
threepoly=sortrows(f,2);
fp2=[peaks2; ptime2];
fl2=[lows2; ltime2];
f2=[fp2 fl2];
f2=f2';
twopoly=sortrows(f2,2);
fp4=[peaks4; ptime4];
fl4=[lows4; ltime4];
f4=[fp4 fl4];
f4=f4';
fourpoly=sortrows(f4,2);
 th=threepoly(:,2);
 th2=diff(th);
 flap_freq=omg_rad_sec*2*abs(mean(th2))/(2*pi);
 tw=twopoly(:,2);
 tw2=diff(tw);
 tw3=omg_rad_sec*2*abs(mean(tw2))/(2*pi);
 fo=fourpoly(:,2);
 fo2=diff(fo);
 feather_freq=omg_rad_sec*2*abs(mean(fo2))/(2*pi);
%_plots_____________________________________________________________
figure(2)
hold
plot(time,bb_dis_3, 'k')
xlabel('time (s)')
ylabel('Flapping amplitude at blade tip (ft)')
```

```
ylim([min(bb_dis_3) max(bb_dis_3)])
figure(3)
hold
plot(time,bb_dis_2, 'k')
xlabel('time (s)')
ylabel('Lead-Lag amplitude at blade tip (ft)')
ylim([min(bb_dis_2) max(bb_dis_2)])
figure(4)
hold
plot(time,bb_dis_4, 'k')
xlabel('time (s)')
ylabel('Twist amplitude at blade tip (rad)')
ylim([min(bb_dis_4) max(bb_dis_4)])
 p=polyfit(ptime,peaks,3);
 p2=polyfit(ptime2,peaks2,length(ptime2));
 p4=polyfit(ptime4,peaks4,3);
  l=polyfit(ltime,lows,2);
 l2=polyfit(ltime2,lows2,length(ltime2));
 l4=polyfit(ltime4,lows4,2);
max_amp=max(peaks)-min(lows);
max_amp2=max(peaks2)-min(lows2);
max_amp4=max(peaks4)-min(lows4);
fark=polyval(p,time)-polyval(l,time);
fark2=polyval(p2,time)-polyval(l2,time);
fark4=polyval(p4,time)-polyval(l4,time);
tim=find(fark<=max_amp*0.5);
tim2=find(fark2<=max_amp2*0.5);
tim4=find(fark4<=max_amp4*0.5);
if isempty(tim)==true
    ti=1;
else
    ti=tim(1);
end
t_stab=time(ti);
if isempty(tim2)==true
    ti2=1;
else
    ti2=tim2(1);
end
t_stab2=time(ti2);
if isempty(tim4)==true
    ti4=1;
else
    ti4=tim4(1);
end
t_stab4=time(ti4);
dynamic_stability=(2*t_stab+t_stab2+t_stab4+2*max_amp+max_amp2+max_amp4)
else
dynamic_stability=1000;
flap_freq=0;
feather_freq=0;
end
```

E.2 Some of the ANSYS Macros Used in Modelcenter

Those macros were used for generating the 2D ANSYS geometry by using the information from an Excel sheet. The Excel sheet information is first fed into ModelCenter, then the following files were generated for each flexure crossection.

E.2.1 flex1.mac

```
/BATCH
/input,menust,tmp,'',,,,,,,,,,,,,,,1
```

```
!/GRA,POWER
/GST,ON
/PLO,INFO,3
/GRO,CURL,ON
/CPLANE,1
/REPLOT,RESIZE
WPSTYLE,,,,,,,,0
/CWD,F:\ModelCenterFolder\KU_haiying\ANSYS
!flex1
geometr,0.0586018985030601,0.0760154818363935,0.0188310416666667,0.0188310416666667,
0.109271061950805,0.126684645284139
vabsinp,'flex_cross1'
/sys, echo flex_cross1.dat | vabs
finish
```

E.2.2 flex2.mac

```
/BATCH
/input,menust,tmp,'',,,,,,,,,,,,,,,1
!/GRA,POWER
/GST,ON
/PLO,INFO,3
/GRO,CURL,ON
/CPLANE,1
/REPLOT,RESIZE
WPSTYLE,,,,,,,,0
/CWD,F:\ModelCenterFolder\KU_haiying\ANSYS
!flex2
geometr,0.0528396547807651,0.0702532381140984,0.0188310416666667,0.0188310416666667,
0.100788247343587,0.11820183067692
vabsinp,'flex_cross2'
/sys, echo flex_cross2.dat | vabs

finish
```

E.2.3 flex3.mac

```
/BATCH
/input,menust,tmp,'',,,,,,,,,,,,,,,1
!/GRA,POWER
/GST,ON
/PLO,INFO,3
/GRO,CURL,ON
/CPLANE,1
/REPLOT,RESIZE
WPSTYLE,,,,,,,,0
/CWD,F:\ModelCenterFolder\KU_haiying\ANSYS
!flex3
geometr,0.0499526449402467,0.06736622827358,0.0188310416666667,0.0188310416666667,
0.0965429681705019,0.113956551503835
vabsinp,'flex_cross3'
/sys, echo flex_cross3.dat | vabs

finish
```

E.2.4 flex4.mac

```
/BATCH
/input,menust,tmp,'',,,,,,,,,,,,,,,1
!/GRA,POWER
/GST,ON
/PLO,INFO,3
/GRO,CURL,ON
/CPLANE,1
/REPLOT,RESIZE
WPSTYLE,,,,,,,,0
```

```
/CWD,F:\ModelCenterFolder\KU_haiying\ANSYS
!flex4
geometr,0.0441636342445532,0.0615772175778866,0.0188310416666667,0.0188310416666667,
0.0880425942651325,0.105456177598466
vabsinp,'flex_cross4'
/sys, echo flex_cross4.dat | vabs
finish
```

E.2.5 flex5.mac

```
/BATCH
/input,menust,tmp,'',,,,,,,,,,,,,,,1
!/GRA,POWER
/GST,ON
/PLO,INFO,3
/GRO,CURL,ON
/CPLANE,1
/REPLOT,RESIZE
WPSTYLE,,,,,,,,0
/CWD,F:\ModelCenterFolder\KU_haiying\ANSYS
!flex5
geometr,0.0205906354390317,0.0380042187723651,0.0188310416666667,0.0188310416666667,
0.0537744806170912,0.0711880639504246
vabsinp,'flex_cross5'
/sys, echo flex_cross5.dat | vabs
finish
```

E.2.6 flex6.mac

```
/BATCH
/input,menust,tmp,'',,,,,,,,,,,,,,,1
!/GRA,POWER
/GST,ON
/PLO,INFO,3
/GRO,CURL,ON
/CPLANE,1
/REPLOT,RESIZE
WPSTYLE,,,,,,,,0
/CWD,F:\ModelCenterFolder\KU_haiying\ANSYS
!flex6
geometr,0.0239081594772218,0.0413217428105551,0.0188310416666667,0.0188310416666667,
0.0413217428105551,0.0587353261438884
vabsinp,'flex_cross6'
/sys, echo flex_cross6.dat | vabs
finish
```

E.2.7 geometr.mac

```
/prep7
!-------------------------------
!      Keypoints
!-------------------------------
!Horizontal flexures (on the sides)
!-------------------------------
y_hor1=arg1
y_hor2=arg2
z_hor1=arg3
!-------------------------------

K,1,0,y_hor2,z_hor1
K,2,0,y_hor1,z_hor1
KSYMM,Z,1      !K3
KSYMM,Y,1,3,2    !K5

!-------------------------------
!Vertical flexures (at the top and bottom)
!-------------------------------
y_ver1=arg4
z_ver1=arg5
```

```
z_ver2=arg6
!------------------------------

K,9,0,y_ver1,z_ver1
K,10,0,y_ver1,z_ver2
KSYMM,Y,10      !K6
KSYMM,Z,6,10,4   !K8

!------------------------------
!     Lines
!------------------------------

L,1,2
L,9,10
L2TAN,1,2 !tangent to lines 1 and 2
LSYMM,y,3 !Y symmetry, line 3       K12
LSYMM,z,3,4 !Z symmetry, lines 3 to 4 K16

!------------------------------
!     Areas
!------------------------------

A,1,2,13,3
A,9,10,6,12
A,4,5,15,11
A,7,8,14,16
AL,7,3,12,4,15,6,19,5
```

E.3 Wrapper Files Used in ModelCenter

E.3.1 ANSYS Wrapper

```
 #@author: Emre Gunduz, HaiYing Liu
 #@version: May 2009
 #@description: ANSYS batch mode input file

#----------------------#
#  Input File Section  #
#----------------------#

#=============================================================================0
RowFieldInputFile dummy
{
   templateFile: dummy.template
   initializationFile: dummy.template
   fileToGenerate: dummy.txt
   #          Name    Type    Row    Field
   #-------  ------  ------  -----  -------

   setDelimiters ","
#            Name      Type    row     field
# ------------------------------------------
   variable: static_stability  double  1  1

clearMarks
}

#=============================================================================1
RowFieldInputFile flex_one
{
   templateFile: flex1.template
   initializationFile: flex1.template
   fileToGenerate: flex1.mac
   #          Name    Type    Row    Field
   #-------  ------  ------  -----  -------
   markAsBeginning "!flex1"
   setDelimiters ","
#            Name      Type    row     field
# ------------------------------------------
   variable: horizontal_flex_cross_1_y1  double  2  2
 variable: horizontal_flex_cross_1_y2  double  2  3
 variable: horizontal_flex_cross_1_z1  double  2  4
 variable: vertical_flex_cross_1_y1  double  2  5
 variable: vertical_flex_cross_1_z1  double  2  6
 variable: vertical_flex_cross_1_z2  double  2  7
 #variable: eta1  double  3  3
clearMarks
```

```
}
#=============================================================================2
RowFieldInputFile flex_two
{
   templateFile: flex2.template
   initializationFile: flex2.template
   fileToGenerate: flex2.mac
   #          Name    Type    Row    Field
   #-------  ------  ------  -----  -------
   markAsBeginning "!flex2"
   setDelimiters ","
#           Name      Type    row     field
# ------------------------------------------
   variable: horizontal_flex_cross_2_y1  double  2  2
 variable: horizontal_flex_cross_2_y2  double  2  3
 variable: horizontal_flex_cross_2_z1  double  2  4
 variable: vertical_flex_cross_2_y1  double  2  5
 variable: vertical_flex_cross_2_z1  double  2  6
 variable: vertical_flex_cross_2_z2  double  2  7
 #variable: eta2  double  3  3
clearMarks
}
#=============================================================================3
RowFieldInputFile flex_three
{
   templateFile: flex3.template
   initializationFile: flex3.template
   fileToGenerate: flex3.mac
#          Name    Type    Row    Field
   #-------  ------  ------  -----  -------
   markAsBeginning "!flex3"
   setDelimiters ","
#           Name      Type    row     field
# ------------------------------------------
   variable: horizontal_flex_cross_3_y1  double  2  2
 variable: horizontal_flex_cross_3_y2  double  2  3
 variable: horizontal_flex_cross_3_z1  double  2  4
 variable: vertical_flex_cross_3_y1  double  2  5
 variable: vertical_flex_cross_3_z1  double  2  6
 variable: vertical_flex_cross_3_z2  double  2  7
 #variable: eta3  double  3  3
clearMarks
}
#=============================================================================4
RowFieldInputFile flex_four
{
   templateFile: flex4.template
   initializationFile: flex4.template
   fileToGenerate: flex4.mac
#          Name    Type    Row    Field
   #-------  ------  ------  -----  -------
   markAsBeginning "!flex4"
   setDelimiters ","
#           Name      Type    row     field
# ------------------------------------------
   variable: horizontal_flex_cross_4_y1  double  2  2
 variable: horizontal_flex_cross_4_y2  double  2  3
 variable: horizontal_flex_cross_4_z1  double  2  4
 variable: vertical_flex_cross_4_y1  double  2  5
 variable: vertical_flex_cross_4_z1  double  2  6
 variable: vertical_flex_cross_4_z2  double  2  7
 #variable: eta4  double  3  3
clearMarks
}
#=============================================================================5
RowFieldInputFile flex_five
{
   templateFile: flex5.template
   initializationFile: flex5.template
   fileToGenerate: flex5.mac
#          Name    Type    Row    Field
   #-------  ------  ------  -----  -------
   markAsBeginning "!flex5"
   setDelimiters ","
#           Name      Type    row     field
# ------------------------------------------
```

```
   variable: horizontal_flex_cross_5_y1  double  2  2
 variable: horizontal_flex_cross_5_y2  double  2  3
 variable: horizontal_flex_cross_5_z1  double  2  4
 variable: vertical_flex_cross_5_y1  double  2  5
 variable: vertical_flex_cross_5_z1  double  2  6
 variable: vertical_flex_cross_5_z2  double  2  7
 #variable: eta5  double  3  3
clearMarks
}
#============================================================================6
RowFieldInputFile flex_six
{
   templateFile: flex6.template
   initializationFile: flex6.template
   fileToGenerate: flex6.mac
#          Name    Type    Row    Field
   #-------  ------  ------  -----  -------
   markAsBeginning "!flex6"
   setDelimiters ","
#           Name      Type    row     field
# ------------------------------------------
   variable: horizontal_flex_cross_6_y1  double  2  2
 variable: horizontal_flex_cross_6_y2  double  2  3
 variable: horizontal_flex_cross_6_z1  double  2  4
 variable: vertical_flex_cross_6_y1  double  2  5
 variable: vertical_flex_cross_6_z1  double  2  6
 variable: vertical_flex_cross_6_z2  double  2  7
 #variable: eta6  double  3  3
clearMarks
}
#============================================================================
#============================================================================

RunCommands
{
  run "clean.bat"
  generate flex_one
  run "'C:\\Program Files\\ANSYS Inc\\v110\\ANSYS\\bin\\intel\\ansys110.exe'
  -p AA_T_A -dir   F:\\ModelCenterFolder\\KU_haiying\\ANSYS -j batch_macros
  -s read -l en-us -b -i flex1.mac -o flex1.out"
  parse outputFlex1

  generate flex_two
  run "'C:\\Program Files\\ANSYS Inc\\v110\\ANSYS\\bin\\intel\\ansys110.exe'
  -p AA_T_A -dir F:\\ModelCenterFolder\\KU_haiying\\ANSYS -j batch_macros
  -s read -l en-us -b -i flex2.mac -o flex2.out"
  parse outputFlex2

  generate flex_three
run "'C:\\Program Files\\ANSYS Inc\\v110\\ANSYS\\bin\\intel\\ansys110.exe'
-p AA_T_A -dir F:\\ModelCenterFolder\\KU_haiying\\ANSYS -j batch_macros
-s read -l en-us -b -i flex3.mac -o flex3.out"
  parse outputFlex3

  generate flex_four
run "'C:\\Program Files\\ANSYS Inc\\v110\\ANSYS\\bin\\intel\\ansys110.exe'
-p AA_T_A -dir   F:\\ModelCenterFolder\\KU_haiying\\ANSYS -j batch_macros
-s read -l en-us -b -i flex4.mac -o flex4.out"
  parse outputFlex4

  generate flex_five
run "'C:\\Program Files\\ANSYS Inc\\v110\\ANSYS\\bin\\intel\\ansys110.exe'
-p AA_T_A -dir F:\\ModelCenterFolder\\KU_haiying\\ANSYS -j batch_macros
-s read -l en-us -b -i flex5.mac -o flex5.out"
  parse outputFlex5

  generate flex_six
run "'C:\\Program Files\\ANSYS Inc\\v110\\ANSYS\\bin\\intel\\ansys110.exe'
-p AA_T_A -dir F:\\ModelCenterFolder\\KU_haiying\\ANSYS -j batch_macros
-s read -l en-us -b -i flex6.mac -o flex6.out"
  parse outputFlex6
  }

#----------------------#
#  Output File Section  #
#----------------------#
```

```
#=========================================================================1
RowFieldOutputFile outputFlex1
{
   fileToParse: flex_cross1.dat.K
   setGroup "PropFlex1"
#  Output: cross sectional properties
   markAsBeginning "THE 6X6 MASS MATRIX (ACCORDING TO DYMORE CONVENTION)"
   setDelimiters " "
#            Name      Type    row     field
# ------------------------------------------
   variable: m0_1        double   4      1   description="mass per span"
   units="slug/ft"
   variable: m11_1       double   7      4   description="polar mass moment of
   inertia per span" units="slug*ft"
   variable: m22_1       double   8      5   description="flapwise mass moment
   of inertia per span" units="slug*ft"
   variable: m33_1       double   9      6   description="edgewise mass moment
   of inertia per span" units="slug*ft"
   clearMarks
   markAsBeginning "THE MASS CENTER"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xm_1        double  4:5     2
  clearMarks
   markAsBeginning "THE CENTROID OF THE CROSS SECTION (PURELY GEOMETRIC)"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xc_1        double  4:5     2
  clearMarks
   markAsBeginning "THE SHEAR CENTER OF THE CROSS SECTION"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xk_1        double  3:4     2
  clearMarks
#  classical beam model: Timoshenko beam
   markAsBeginning "TIMOSHENKO STIFFNESS MATRIX"
   setDelimiters " "
#            Name      Type    row     field
# ------------------------------------------
   variable: EAt_1       double   5      1   description="axial stiffness" units="lbf"
   variable: K22t_1      double   6      2   description="edgewise shearing
   stiffness" units="lbf*ft"
   variable: K33t_1      double   7      3   description="flapwise shearing
   stiffness" units="lbf*ft"
   variable: K23t_1      double   7      2   description="coupled shearing
   stiffness" units="lbf*ft"
   variable: GJt_1       double   8      4   description="torsional stiffness"
   units="lbf*ft"
   variable: EI22t_1     double   9      5   description="flapwise bending
   stiffness" units="lbf*ft"
   variable: EI33t_1     double   10     6   description="edgewise bending
   stiffness" units="lbf*ft"
   variable: EI23t_1     double   10     5   description="coupled bending
   stiffness" units="lbf*ft"
   clearMarks
}
#=========================================================================2
RowFieldOutputFile outputFlex2
{
   fileToParse: flex_cross2.dat.K
   setGroup "PropFlex2"
#  Output: cross sectional properties
   markAsBeginning "THE 6X6 MASS MATRIX (ACCORDING TO DYMORE CONVENTION)"
   setDelimiters " "
#            Name      Type    row     field
# ------------------------------------------
   variable: m0_2        double   4      1   description="mass per span" units="slug/ft"
   variable: m11_2       double   7      4   description="polar mass moment of
   inertia per span" units="slug*ft"
```

```
   variable: m22_2       double   8      5   description="flapwise mass moment of
   inertia per span" units="slug*ft"
   variable: m33_2       double   9      6   description="edgewise mass moment of
   inertia per span" units="slug*ft"
   clearMarks

   markAsBeginning "THE MASS CENTER"
   setDelimiters "="
#            Name      Type    row     field
# -----------------------------------------
   array:    xm_2        double  4:5     2
  clearMarks

   markAsBeginning "THE CENTROID OF THE CROSS SECTION (PURELY GEOMETRIC)"
   setDelimiters "="
#            Name      Type    row     field
# -----------------------------------------
   array:    xc_2        double  4:5     2
  clearMarks

   markAsBeginning "THE SHEAR CENTER OF THE CROSS SECTION"
   setDelimiters "="
#            Name      Type    row     field
# -----------------------------------------
   array:    xk_2        double  3:4     2
  clearMarks

#  classical beam model: Timoshenko beam
   markAsBeginning "TIMOSHENKO STIFFNESS MATRIX"
   setDelimiters " "
#            Name      Type    row     field
# -----------------------------------------
   variable: EAt_2       double   5      1   description="axial stiffness" units="lbf"
   variable: K22t_2      double   6      2   description="edgewise shearing
   stiffness" units="lbf*ft"
   variable: K33t_2      double   7      3   description="flapwise shearing
   stiffness" units="lbf*ft"
   variable: K23t_2      double   7      2   description="coupled shearing
   stiffness" units="lbf*ft"
   variable: GJt_2       double   8      4   description="torsional stiffness"
   units="lbf*ft"
   variable: EI22t_2     double   9      5   description="flapwise bending
   stiffness" units="lbf*ft"
   variable: EI33t_2     double   10     6   description="edgewise bending
   stiffness" units="lbf*ft"
   variable: EI23t_2     double   10     5   description="coupled bending
   stiffness" units="lbf*ft"
   clearMarks
}
#=========================================================================3
RowFieldOutputFile outputFlex3
{
   fileToParse: flex_cross3.dat.K
   setGroup "PropFlex3"
#  Output: cross sectional properties
   markAsBeginning "THE 6X6 MASS MATRIX (ACCORDING TO DYMORE CONVENTION)"
   setDelimiters " "
#            Name      Type    row     field
# -----------------------------------------
   variable: m0_3        double   4      1   description="mass per span" units="slug/ft"
   variable: m11_3       double   7      4   description="polar mass moment of
   inertia per span" units="slug*ft"
   variable: m22_3       double   8      5   description="flapwise mass moment of
   inertia per span" units="slug*ft"
   variable: m33_3       double   9      6   description="edgewise mass moment of
   inertia per span" units="slug*ft"
   clearMarks

   markAsBeginning "THE MASS CENTER"
   setDelimiters "="
#            Name      Type    row     field
# -----------------------------------------
   array:    xm_3        double  4:5     2
  clearMarks

   markAsBeginning "THE CENTROID OF THE CROSS SECTION (PURELY GEOMETRIC)"
   setDelimiters "="
#            Name      Type    row     field
```

```
# ------------------------------------------
   array:    xc_3        double  4:5     2
  clearMarks
   markAsBeginning "THE SHEAR CENTER OF THE CROSS SECTION"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xk_3        double  3:4     2
  clearMarks
#  classical beam model: Timoshenko beam
   markAsBeginning "TIMOSHENKO STIFFNESS MATRIX"
   setDelimiters " "
#            Name      Type    row     field
# ------------------------------------------
   variable: EAt_3       double   5      1   description="axial stiffness" units="lbf"
   variable: K22t_3      double   6      2   description="edgewise shearing
   stiffness" units="lbf*ft"
   variable: K33t_3      double   7      3   description="flapwise shearing
   stiffness" units="lbf*ft"
   variable: K23t_3      double   7      2   description="coupled shearing
   stiffness" units="lbf*ft"
   variable: GJt_3       double   8      4   description="torsional stiffness"
   units="lbf*ft"
   variable: EI22t_3     double   9      5   description="flapwise bending
   stiffness" units="lbf*ft"
   variable: EI33t_3     double   10     6   description="edgewise bending
   stiffness" units="lbf*ft"
   variable: EI23t_3     double   10     5   description="coupled bending
   stiffness" units="lbf*ft"
   clearMarks
}
#=============================================================================4
RowFieldOutputFile outputFlex4
{
   fileToParse: flex_cross4.dat.K
   setGroup "PropFlex4"
#  Output: cross sectional properties
   markAsBeginning "THE 6X6 MASS MATRIX (ACCORDING TO DYMORE CONVENTION)"
   setDelimiters " "
#            Name      Type    row     field
# ------------------------------------------
   variable: m0_4        double   4      1   description="mass per span" units="slug/ft"
   variable: m11_4       double   7      4   description="polar mass moment of
   inertia per span" units="slug*ft"
   variable: m22_4       double   8      5   description="flapwise mass moment of
   inertia per span" units="slug*ft"
   variable: m33_4       double   9      6   description="edgewise mass moment of
   inertia per span" units="slug*ft"
   clearMarks
   markAsBeginning "THE MASS CENTER"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xm_4        double  4:5     2
  clearMarks
   markAsBeginning "THE CENTROID OF THE CROSS SECTION (PURELY GEOMETRIC)"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xc_4        double  4:5     2
  clearMarks
   markAsBeginning "THE SHEAR CENTER OF THE CROSS SECTION"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xk_4        double  3:4     2
  clearMarks
#  classical beam model: Timoshenko beam
   markAsBeginning "TIMOSHENKO STIFFNESS MATRIX"
   setDelimiters " "
#            Name      Type    row     field
# ------------------------------------------
   variable: EAt_4       double   5      1   description="axial stiffness" units="lbf"
```

```
   variable: K22t_4      double   6      2   description="edgewise shearing
   stiffness" units="lbf*ft"
   variable: K33t_4      double   7      3   description="flapwise shearing
   stiffness" units="lbf*ft"
   variable: K23t_4      double   7      2   description="coupled shearing
   stiffness" units="lbf*ft"
   variable: GJt_4       double   8      4   description="torsional stiffness"
   units="lbf*ft"
   variable: EI22t_4     double   9      5   description="flapwise bending
   stiffness" units="lbf*ft"
   variable: EI33t_4     double   10     6   description="edgewise bending
   stiffness" units="lbf*ft"
   variable: EI23t_4     double   10     5   description="coupled bending
   stiffness" units="lbf*ft"
   clearMarks
}
#========================================================================5
RowFieldOutputFile outputFlex5
{
   fileToParse: flex_cross5.dat.K
   setGroup "PropFlex5"
#  Output: cross sectional properties
   markAsBeginning "THE 6X6 MASS MATRIX (ACCORDING TO DYMORE CONVENTION)"
   setDelimiters " "
#            Name      Type    row     field
# -------------------------------------------
   variable: m0_5        double   4      1   description="mass per span" units="slug/ft"
   variable: m11_5       double   7      4   description="polar mass moment of
   inertia per span" units="slug*ft"
   variable: m22_5       double   8      5   description="flapwise mass moment of
    inertia per span" units="slug*ft"
   variable: m33_5       double   9      6   description="edgewise mass moment of
   inertia per span" units="slug*ft"
   clearMarks

   markAsBeginning "THE MASS CENTER"
   setDelimiters "="
#            Name      Type    row     field
# -------------------------------------------
   array:    xm_5        double  4:5     2
  clearMarks

   markAsBeginning "THE CENTROID OF THE CROSS SECTION (PURELY GEOMETRIC)"
   setDelimiters "="
#            Name      Type    row     field
# -------------------------------------------
   array:    xc_5        double  4:5     2
  clearMarks

   markAsBeginning "THE SHEAR CENTER OF THE CROSS SECTION"
   setDelimiters "="
#            Name      Type    row     field
# -------------------------------------------
   array:    xk_5        double  3:4     2
  clearMarks

#  classical beam model: Timoshenko beam
   markAsBeginning "TIMOSHENKO STIFFNESS MATRIX"
   setDelimiters " "
#            Name      Type    row     field
# -------------------------------------------
   variable: EAt_5       double   5      1   description="axial stiffness" units="lbf"
   variable: K22t_5      double   6      2   description="edgewise shearing
   stiffness" units="lbf*ft"
   variable: K33t_5      double   7      3   description="flapwise shearing
   stiffness" units="lbf*ft"
   variable: K23t_5      double   7      2   description="coupled shearing
   stiffness" units="lbf*ft"
   variable: GJt_5       double   8      4   description="torsional stiffness"
    units="lbf*ft"
   variable: EI22t_5     double   9      5   description="flapwise bending
   stiffness" units="lbf*ft"
   variable: EI33t_5     double   10     6   description="edgewise bending
   stiffness" units="lbf*ft"
   variable: EI23t_5     double   10     5   description="coupled bending
   stiffness" units="lbf*ft"
   clearMarks
}
```

```
#=========================================================================6
RowFieldOutputFile outputFlex6
{
   fileToParse: flex_cross6.dat.K
   setGroup "PropFlex6"
#  Output: cross sectional properties
   markAsBeginning "THE 6X6 MASS MATRIX (ACCORDING TO DYMORE CONVENTION)"
   setDelimiters " "
#            Name      Type    row     field
# ------------------------------------------
   variable: m0_6        double   4      1   description="mass per span" units="slug/ft"
   variable: m11_6       double   7      4   description="polar mass moment of
   inertia per span" units="slug*ft"
   variable: m22_6       double   8      5   description="flapwise mass moment of
   inertia per span" units="slug*ft"
   variable: m33_6       double   9      6   description="edgewise mass moment of
   inertia per span" units="slug*ft"
   clearMarks
   markAsBeginning "THE MASS CENTER"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xm_6        double  4:5     2
  clearMarks
   markAsBeginning "THE CENTROID OF THE CROSS SECTION (PURELY GEOMETRIC)"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xc_6        double  4:5     2
  clearMarks
   markAsBeginning "THE SHEAR CENTER OF THE CROSS SECTION"
   setDelimiters "="
#            Name      Type    row     field
# ------------------------------------------
   array:    xk_6        double  3:4     2
  clearMarks
#  classical beam model: Timoshenko beam
   markAsBeginning "TIMOSHENKO STIFFNESS MATRIX"
   setDelimiters " "
#            Name      Type    row     field
# ------------------------------------------
   variable: EAt_6       double   5      1   description="axial stiffness" units="lbf"
   variable: K22t_6      double   6      2   description="edgewise shearing
   stiffness" units="lbf*ft"
   variable: K33t_6      double   7      3   description="flapwise shearing
   stiffness" units="lbf*ft"
   variable: K23t_6      double   7      2   description="coupled shearing
   stiffness" units="lbf*ft"
   variable: GJt_6       double   8      4   description="torsional stiffness"
   units="lbf*ft"
   variable: EI22t_6     double   9      5   description="flapwise bending
   stiffness" units="lbf*ft"
   variable: EI33t_6     double   10     6   description="edgewise bending
    stiffness" units="lbf*ft"
   variable: EI23t_6     double   10     5   description="coupled bending
    stiffness" units="lbf*ft"
   clearMarks
}
```

E.3.2 Wrapper for DYMORE Static Analysis

```
#-----------------------------------------------#
#  DYMORE input/output data of static analysis  #
# INPUT data : geometry information             #
#              rotor speed                      #
#              flexure property                 #
# OUTPUT data : frequencies                     #
#-----------------------------------------------#

#-----------------#
#  Header Section  #
#-----------------#
```

```
# @author: Georgia Tech
# @version: version 1
# @description: DYMORE eigen analysis file-wrapper for Rambler Rotor (Hanson idea)
#----------------------#
#  RunCommand Section  #
#----------------------#
userVariable: StaJobFileName string default=LHY_static
userVariable: IniFileName string default=LHY_static
userVariable: PropFileName string default=dimension_prop.dat
RunCommands
{
   generate dimension_prop
   generate StaticInputFile
#   run \${StaJobFileName}.dym
   run "dymore.exe \${StaJobFileName}.dym"
   parse StaticOutputFile
}
#----------------------#
#  Input File Section  #
#----------------------#
RowFieldInputFile dimension_prop
{
#  generate dimenion and property file: dimensionprop.dat
   templateFile: template\dimension_prop.dat.template
   fileToGenerate: model\dimension_prop.dat
#  update points
   markAsBeginning "@POINT_NAME { point_flex_rootQ1Q } {"
   setDelimiters "{ , }"
   #         name          type    row     field
   array: point_flex_root       double  2       2:4
   clearMarks
   markAsBeginning "@POINT_NAME { point_blade_rootQ1Q } {"
   setDelimiters "{ , }"
   #         name          type    row     field
   array: point_blade_root      double  2 2:4
   clearMarks
   markAsBeginning "@POINT_NAME { point_blade_tipQ1Q } {"
   setDelimiters "{ , }"
   #         name          type    row     field
   array: point_blade_tip       double  2 2:4
   clearMarks
   markAsBeginning "@POINT_NAME { point_blade_tipwtQ1Q } {"
   setDelimiters "{ , }"
   #         name          type    row     field
   array: point_blade_tipwt     double  2 2:4
   clearMarks
   markAsBeginning "@POINT_NAME { point_pitchlink_shaftQ1Q } {"
   setDelimiters "{ , }"
   #         name          type    row     field
   array: point_pitchlink_shaft double  2 2:4
   clearMarks
   markAsBeginning "@POINT_NAME { point_pitchlink_tubeQ1Q } {"
   setDelimiters "{ , }"
   #         name          type    row     field
   array: point_pitchlink_tube  double  2 2:4
   clearMarks
#  update tip weight
   markAsBeginning "@MASS_PROPERTY_NAME {property_tipwt} {"
   setDelimiters "{ , }"
   #         name          type    row     field
   variable: tip_weight         double  2 2
   clearMarks
#  update frame systems
   markAsBeginning "@FIXED_FRAME_NAME { FrameBladePreconeQ1Q } {"
```

```
   setDelimiters "{ , }"
   #         name          type    row     field
   array: origin_frame_precone  double  2 2:4
   variable: precone            double  3 3
   clearMarks

   markAsBeginning "@FIXED_FRAME_NAME { FrameBladeSweepQ1Q } {"
   setDelimiters "{ , }"
   #         name          type    row     field
   array: origin_frame_sweep    double  2 2:4
   variable: sweep              double  3 3
   clearMarks

#  update flexure property
   markAsBeginning "@BEAM_PROPERTY_NAME {property_flexbeam} {"
   setDelimiters "{"
   #         name   type     row  field
variable: eta1 double 2  2
variable: EAt_1 double 4  2
variable: GJt_1 double 9 2
variable: m0_1 double 14 2

setDelimiters ","
variable: EI22t_1 double 7 1
variable: EI33t_1 double 7 2
variable: K22t_1 double 12 1
variable: K33t_1 double 12 2
variable: m11_1 double 17 1
variable: m22_1 double 17 2
variable: m33_1 double 17 3

markAsBeginning "! second crossection "
   setDelimiters "{"
   #         name   type     row  field
variable: eta2 double 2  2
variable: EAt_2 double 4  2
variable: GJt_2 double 9 2
variable: m0_2 double 14 2

setDelimiters ","
variable: EI22t_2 double 7 1
variable: EI33t_2 double 7 2
variable: K22t_2 double 12 1
variable: K33t_2 double 12 2
variable: m11_2 double 17 1
variable: m22_2 double 17 2
variable: m33_2 double 17 3

markAsBeginning "! third crossection "
   setDelimiters "{"
   #         name   type     row  field
variable: eta3 double 2  2
variable: EAt_3 double 4  2
variable: GJt_3 double 9 2
variable: m0_3 double 14 2

setDelimiters ","
variable: EI22t_3 double 7 1
variable: EI33t_3 double 7 2
variable: K22t_3 double 12 1
variable: K33t_3 double 12 2
variable: m11_3 double 17 1
variable: m22_3 double 17 2
variable: m33_3 double 17 3

markAsBeginning "! fourth crossection "
   setDelimiters "{"
   #         name   type     row  field
variable: eta4 double 2  2
variable: EAt_4 double 4  2
variable: GJt_4 double 9 2
variable: m0_4 double 14 2

setDelimiters ","
variable: EI22t_4 double 7 1
variable: EI33t_4 double 7 2
variable: K22t_4 double 12 1
variable: K33t_4 double 12 2
variable: m11_4 double 17 1
variable: m22_4 double 17 2
```

```
variable: m33_4 double 17 3

markAsBeginning "! fifth crossection "
   setDelimiters "{"
   #         name   type     row  field
variable: eta5 double 2  2
variable: EAt_5 double 4  2
variable: GJt_5 double 9 2
variable: m0_5 double 14 2

setDelimiters ","
variable: EI22t_5 double 7 1
variable: EI33t_5 double 7 2
variable: K22t_5 double 12 1
variable: K33t_5 double 12 2
variable: m11_5 double 17 1
variable: m22_5 double 17 2
variable: m33_5 double 17 3

markAsBeginning "! sixth crossection "
   setDelimiters "{"
   #         name   type     row  field
variable: eta6 double 2  2
variable: EAt_6 double 4  2
variable: GJt_6 double 9 2
variable: m0_6 double 14 2

setDelimiters ","
variable: EI22t_6 double 7 1
variable: EI33t_6 double 7 2
variable: K22t_6 double 12 1
variable: K33t_6 double 12 2
variable: m11_6 double 17 1
variable: m22_6 double 17 2
variable: m33_6 double 17 3

}

RowFieldInputFile StaticInputFile
{
#  generate static main input file: LHY_static.dym
   templateFile: template\${StaJobFileName}.dym.template
   fileToGenerate: \${StaJobFileName}.dym
#  update rotor speed
   markAsBeginning "@RIGID_ROTATION_DEFINITION {"
   setDelimiters "{ , }"
#            Name       Type   row   field   options
# ------------------------------------------------
   variable: omg_rad_sec   double 6     4       description="rotor speed"
   units="rad/sec"
   clearMarks
}

#-----------------------#
#  Output File Section  #
#-----------------------#

RowFieldOutputFile StaticOutputFile
{
   fileToParse: \${StaJobFileName}.eig
   markAsBeginning "******:" occurrence=-5
   setDelimiters "     "
#         name  type     row  field
   variable: dymore_worked double 1 4

   markAsBeginning "EigenValue" occurrence=1
    setDelimiters "     "
    setGroup "mode1"
   array: x_disp double 2:37 1
array: y_disp double 2:37 2
array: z_disp double 2:37 3
array: x_rot double 2:37 4
array: y_rot double 2:37 5
array: z_rot double 2:37 6

markAsBeginning "EigenValue" occurrence=2
    setDelimiters "     "
   setGroup "mode2"
```

```
   array: x_disp double 2:37 1
array: y_disp double 2:37 2
array: z_disp double 2:37 3
array: x_rot double 2:37 4
array: y_rot double 2:37 5
array: z_rot double 2:37 6
   markAsBeginning "EigenValue" occurrence=3
    setDelimiters "     "
   setGroup "mode3"
   array: x_disp double 2:37 1
array: y_disp double 2:37 2
array: z_disp double 2:37 3
array: x_rot double 2:37 4
array: y_rot double 2:37 5
array: z_rot double 2:37 6
   markAsBeginning "EigenValue" occurrence=4
    setDelimiters "     "
   setGroup "mode4"
   array: x_disp double 2:37 1
array: y_disp double 2:37 2
array: z_disp double 2:37 3
array: x_rot double 2:37 4
array: y_rot double 2:37 5
array: z_rot double 2:37 6
}
```

E.3.3 Wrapper for DYMORE Dynamic Analysis

```
#-----------------------------------------------#
#  DYMORE input/output data of static analysis  #
# INPUT data : geometry information             #
#              rotor speed                      #
#              blade properties                 #
#              blade structural twist           #
# OUTPUT data : frequencies                     #
#-----------------------------------------------#

#-----------------#
#  Header Section  #
#-----------------#
# @author: Georgia Tech
# @version: version 1
# @description: DYMORE file wrapper for Rambler Rotor (1-bladed model) (Hanson idea)

#---------------------#
#  RunCommand Section  #
#---------------------#

userVariable: DynJobFileName string default=LHY_dynamic
userVariable: OutFileName string default=FIGURES\SensBeam1Dis000.mdt

RunCommands
{
   generate AerodynamicFile
   generate DynamicInputFile
#   run ${DynJobFileName}.dym
   run "dymore.exe ${DynJobFileName}.dym"
   parse DynamicOutputFile
}

#---------------------#
#  Input File Section  #
#---------------------#

RowFieldInputFile dummyfile
{
#  generate dynamic main input file: LHY_dynamic.dym
   templateFile: template\dummy.template
   fileToGenerate: dummy.txt

   setDelimiters "{ / }"
#            Name       Type   row   field   options
```

```
# ------------------------------------------------
   variable: dymore_worked  double 1     1
   clearMarks
}
RowFieldInputFile DynamicInputFile
{
#  generate dynamic main input file: LHY_dynamic.dym
   templateFile: template\${DynJobFileName}.dym.template
   fileToGenerate: ${DynJobFileName}.dym
#  update rotor geometry and flexture prop
   markAsBeginning "@INCLUDE_COMMAND_NAME {IncludeRotorDimensionPropertyData} {"
   setDelimiters "{ / }"
#            Name       Type   row   field   options
# ------------------------------------------------
   variable: PropFileName  string 4     2
   clearMarks
#  update rotor speed
   markAsBeginning "@TIME_FUNCTION_NAME {control_rotation_shaft} {"
   setDelimiters "{ , }"
#            Name       Type   row   field   options
# ------------------------------------------------
   variable: final_psi  double 5     4
   clearMarks
#  update initial condition
   markAsBeginning "@INITIAL_CONDITION_DEFINITION {"
   setDelimiters "{ . }"
#            Name       Type   row   field   options
# ------------------------------------------------
   variable: IniFileName  string  3    2
   clearMarks
}
RowFieldInputFile AerodynamicFile
{
#  generate main input file: aerodynamic.dat
   templateFile: template\aerodynamic.dat.template
   fileToGenerate: model\aerodynamic.dat
#  update rotor definition
   markAsBeginning "@ROTOR_DEFINITION {"
   setDelimiters "{ , }"
#            Name        Type    row   field   options
# ------------------------------------------------
   variable: omg_rad_sec double  4     2       description="rotor speed" units="rad/sec"
   variable: radius      double  5     2       description="rotor radius" units="ft"
   variable: radius    double  9  2  description="rotor radius" units="ft"
   variable: chord  double  10    2  description="chord length" units="ft"
   clearMarks
#  update lfnline curvilinear coordinates
   markAsBeginning "@LIFTING_LINE_DEFINITION {"
   setDelimiters "{ , }"
#            Name        Type    row   field   options
# ------------------------------------------------
   variable: lfnSi          double  10     2       description="initial curvilinear
   coordinate" units="ft"
   variable: lfnSf          double  10     3       description="final curvilinear
   coordinate" units="ft"
   clearMarks
markAsBeginning "@LIFTING_LINE_PROPERTY_DEFINITION {"
   setDelimiters "{ , }"
#            Name        Type    row   field   options
# ------------------------------------------------
   variable: chord  double  5    2  description="chord length" units="ft"
   clearMarks
}
#-----------------------#
#  Output File Section  #
#-----------------------#
```

```
RowFieldOutputFile DynamicOutputFile
{
   fileToParse: ${OutFileName}
#  Output: a time history of root force of the flexure beam
   setDelimiters " "
#            Name      Type    row     field
# -----------------------------------------
#   setGroup "TimeHistoryofLoadings"
   array:    time    double  2:-1  1
   array:    LL_disp    double  2:-1  3
   array:    flap_disp    double  2:-1  4
   array:    tw_disp    double  2:-1  5
   variable: dymore_worked_dyn double -1 1
}
```

E.4 DYMORE Input Files for Static Analysis

E.4.1 Main File

```
@PROCESS_CONTROL_DEFINITION {
  @UNIT_SYSTEM {US}
  @DEBUG_PRINT_FLAG {     3}
  @FINITE_ELEMENT_ANALYSIS {YES}
  @POST_PROCESSING_ANALYSIS {YES}
  @SIGNAL_PROCESSING_ANALYSIS {YES}
  @MANUAL_PATH {C:\dymore2_0\manual\}
  @FIGURES_PATH {FIGURES\}
}
@MODEL_DEFINITION {
@FIXED_FRAME_DEFINITION {
  @FIXED_FRAME_NAME {frame_blade1} {
    @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
  }
}
@RIGID_ROTATION_DEFINITION {
  @RIGID_ROTATION_NAME {Rigid_rotation} {
    @CONNECTED_TO_BODY {rvj_shaft}
    @AT_POINT {point_shaft}
    @TIME_FUNCTION_NAME {angular_velocity_schedule}
    @ANGULAR_VELOCITY { 0.0, 0.0,  32.8134}
    @BODY_LIST { flexbeamQ1Q, tipwtQ1Q, bladeQ1Q }
  }
}
@GRAVITY_DEFINITION {
  @GRAVITY_NAME {GravityLoading} {
    @GRAVITY_VECTOR { 0.00000e+000,  0.00000e+000, 0.0}
    !-3.22000e+001}
    @TIME_FUNCTION_NAME {gravity_schedule_sta}
  }
}
@TIME_FUNCTION_DEFINITION {
  @TIME_FUNCTION_NAME {control_rotation_shaft} {
    @TIME_FUNCTION_TYPE {USER_DEFINED}
    @TABLE_ENTRIES {
      @TIME { 0.00000e+000}  @FUNCTION_VALUE { 0.00000e+000}
      @TIME { 1.00000e+002}  @FUNCTION_VALUE { 0.00000e+000}
    }
  }
  @TIME_FUNCTION_NAME {prj_control_input} {
    @TIME_FUNCTION_TYPE {USER_DEFINED}
    @TABLE_ENTRIES {
      @TIME { 0.00000e+000}  @FUNCTION_VALUE { 0.00000e+000}
      @TIME { 1.00000e+002}  @FUNCTION_VALUE { 0.00000e+000}
    }
```

```
  }
  @TIME_FUNCTION_NAME {gravity_schedule_sta} {
    @TIME_FUNCTION_TYPE {USER_DEFINED}
    @TABLE_ENTRIES {
      @TIME { 0.00000e+000}  @FUNCTION_VALUE { 0.00000e+000}
      @TIME { 1.00000e-001}  @FUNCTION_VALUE { 0.00000e+000}
      @TIME { 1.00000e+000}  @FUNCTION_VALUE { 1.00000e-001}
      @TIME { 6.00000e+000}  @FUNCTION_VALUE { 1.00000e+000}
      @TIME { 1.00000e+002}  @FUNCTION_VALUE { 1.00000e+000}
    }
  }
  @TIME_FUNCTION_NAME {angular_velocity_schedule} {
    @TIME_FUNCTION_TYPE {USER_DEFINED}
    @TABLE_ENTRIES {
      @TIME { 0.00000e+000}  @FUNCTION_VALUE { 0.00000e+000}
      @TIME { 1.00000e+000}  @FUNCTION_VALUE { 1.00000e-001}
      @TIME { 2.00000e+000}  @FUNCTION_VALUE { 2.00000e-001}
      @TIME { 3.00000e+000}  @FUNCTION_VALUE { 4.00000e-001}
      @TIME { 4.00000e+000}  @FUNCTION_VALUE { 6.00000e-001}
      @TIME { 5.00000e+000}  @FUNCTION_VALUE { 8.00000e-001}
      @TIME { 6.00000e+000}  @FUNCTION_VALUE { 1.00000e+000}
      @TIME { 7.00000e+000}  @FUNCTION_VALUE { 1.20000e+000}
      @TIME { 8.00000e+000}  @FUNCTION_VALUE { 1.20000e+000}
      @TIME { 1.00000e+002}  @FUNCTION_VALUE { 1.20000e+000}
    }
  }
}
@SENSOR_DEFINITION {
  @SENSOR_NAME {SensorEnergy} {
    @OBJECT_NAME {ENERGY}
  }
  @SENSOR_NAME {SensBeam1Dis10} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {DISPLACEMENTS}
    @ETA_VALUE { 1.00000e+000}
  }
}
@INCLUDE_COMMAND {
  @INCLUDE_COMMAND_NAME {IncludeRotorDimensionPropertyData} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/dimension_prop.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeBladeProperties} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/properties.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeSensorsEigen} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/sensors_static_eigen.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludePoint} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/points.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeHub} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/hub.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeFlexBlade} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/flexblade.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeTorqueTube} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
```

```
    model/torquetube.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludePitchLink} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/pitchlink.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeShape} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/shapes.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeGrfParameters} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/grfparameters.dat}
  }
}
}
@CREATE_FINITE_ELEMENT_MODEL {
@CFM_CONTROL_PARAMETERS {
  @ANALYSIS_TYPE {STATIC}
}
}
@FINITE_ELEMENT_ANALYSIS {
@FEM_CONTROL_PARAMETERS {
  @FEM_CONTROL_PARAMETERS_NAME {AnalysisControlParameters} {
    @MAXIMUM_NUMBER_OF_TIME_STEPS {    124}
    @SIMULATION_TIME_RANGE { 0.00000e+000,  6.11e+000}
    @TIME_STEP_SIZE_RANGE { 1.00000e-003,  1.00000e-001}
    @REFERENCE_ENERGY_LEVEL { 3.00000e+007}
  }
}
@INITIAL_CONDITION_DEFINITION {
}
@STEP_CONTROL_PARAMETERS {
  @STEP_CONTROL_PARAMETER_NAME {Step1} {
    @ARCHIVAL_FREQUENCY {     1}
    @REUSE_NUMBER {100000}
    @NUMBER_OF_EIGENVALUES {    5}
    @EIGENPROBLEM_PRINT_FLAG {     0}
    @GYROSCOPIC_TERMS {NO}
    @EIGEN_SPECTRUM_SHIFT { 0.00000e+000}
    @SPECTRAL_RADIUS_AT_INFINITY { 0.00000e+000}
    @TIME_STEP_SIZE { 1.00000e-001}
    @MAXIMUM_NUMBER_OF_ITERATIONS {    10}
    @FACTORIZATION_STRATEGY {     1}
    @CONVERGENCE_TOLERANCE { 1.00000e-0012}
    @MAXIMUM_NUMBER_OF_REJECT {     5}
    @AVERAGE_STIFFNESS_TERM { 1.00000e+006}
    @AVERAGE_MASS_TERM { 0.00000e+000}
  }
}
}
@POST_PROCESSING_ANALYSIS {
@GRAPHICS_CONTROL_PARAMETERS {
  @GRAPHICS_CONTROL_PARAMETERS_NAME {GraphicsControlParameters} {
    @TIME_STEP_SIZE { 0.00000e+000}
    @EIGENVECTORS_SCALING_FACTOR { 2.00000e-001}
    @MODAL_ANIMATION_CYCLES {     5}
    @MODAL_ANIMATION_FRAMES_PER_CYCLE {    50}
    @VECTOR_FIELD_TYPE {FORCES}
    @VECTOR_FIELD_SCALING_FACTOR { 2.00000e-001}
    @VECTOR_FIELD_TYPE {VELOCITIES}
    @VECTOR_FIELD_SCALING_FACTOR { 2.00000e-001}
  }
}
@VIEW_PARAMETERS_DEFINITION {
  @VIEW_PARAMETERS_NAME {ViewParameters} {
    @VIEW_REFERENCE_POINT { 5.00000e-001,  5.00000e-001,  5.00000e-001}
    @VIEW_SIZE { 1.20000e+000,  1.20000e+000,  1.20000e+000}
```

```
    @PROJECTION_REFERENCE_POINT {-8.15283e-001, -8.24033e-001, -9.71433e-001}
    @PROJECTION_EYE_VECTOR { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @PROJECTION_UP_VECTOR { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @PROJECTION_VIEWPORT {-3.71702e-001,  3.71702e-001, -2.55415e-001,  2.55415e-001}
  }
}
}
```

E.4.2 dimension_prop.dat

```
@POINT_DEFINITION {
  @POINT_NAME { point_flex_rootQ1Q } {
    @COORDINATES { 0.565710802223971,   0.0333333333333333,   0.0}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
  @POINT_NAME { point_blade_rootQ1Q } {
    @COORDINATES { 1.700076,   0.0,   0.0}
    @IS_DEFINED_IN_FRAME {FrameBladePreconeQ1Q}
  }
  @POINT_NAME { point_blade_tipQ1Q } {
    @COORDINATES { 11.33384,   0.0,   0.0}
    @IS_DEFINED_IN_FRAME {FrameBladeSweepQ1Q}
  }
  @POINT_NAME { point_blade_tipwtQ1Q } {
    @COORDINATES { 11.2205016,   0.0,   0.0}
    @IS_DEFINED_IN_FRAME {FrameBladeSweepQ1Q}
  }
  @POINT_NAME { point_pitchlink_shaftQ1Q } {
    @COORDINATES { 0.565710802223971,   0.0225208,  -1.0}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
  @POINT_NAME { point_pitchlink_tubeQ1Q } {
    @COORDINATES { 0.565710802223971,   0.0225208,   0.0}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
}
@MASS_PROPERTY_DEFINITION {
  @MASS_PROPERTY_NAME {property_tipwt} {
    @TOTAL_MASS { 0.0326307453416149 }
    @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  5.52822e-002,  0.00000e+000}
    @MOMENTS_OF_INERTIA { 1.00000e-003,  0.00000e+000,  0.00000e+000,
    1.00000e-003,  0.00000e+000,  1.00000e-003}
  }
}
@FIXED_FRAME_DEFINITION {
  @FIXED_FRAME_NAME { FrameBladePreconeQ1Q } {
    @ORIGIN {  0.565710802223971,   0.0333333333333333,   0.0}
    @EULER_ANGLES_321 { 0.00000e+000, -5.82980997880845,  0.00000e+000}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
  @FIXED_FRAME_NAME { FrameBladeSweepQ1Q } {
    @ORIGIN {  1.700076,   0.0333333333333333,   0.0}
    @EULER_ANGLES_321 {  2.00000e+000, 2.6487953460457,  0.00000e+000}
    @IS_DEFINED_IN_FRAME {FrameBladePreconeQ1Q}
  }
}
@BEAM_PROPERTY_DEFINITION {
  @BEAM_PROPERTY_NAME {property_flexbeam} {
    @ETA_VALUE {0.0
    }{
      @AXIAL_STIFFNESS {2996152.8
      }
      @BENDING_STIFFNESSES {
18304.578,5873.1484,  0.00000e+000
      }
      @TORSIONAL_STIFFNESS {393.95021
```

```
      }
      @SHEARING_STIFFNESSES {
171464.25,239608.09,  0.00000e+000
      }
      @MASS_PER_UNIT_SPAN {1.03362
      }
      @MOMENTS_OF_INERTIA {
0.00439457,0.00351348,8.81088E-4
      }
      @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
      @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
      @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
      @DAMPING_COEFFICIENT { 1.00000e-002}
          }
  ! second crossection
    @ETA_VALUE {0.0833333333333334
    }{
      @AXIAL_STIFFNESS {2913930.2
      }
      @BENDING_STIFFNESSES {
15625.997,4899.5527,  0.00000e+000
      }
      @TORSIONAL_STIFFNESS {319.89747
      }
      @SHEARING_STIFFNESSES {
155743.05,216184.22,  0.00000e+000
      }
      @MASS_PER_UNIT_SPAN {0.921906
      }
      @MOMENTS_OF_INERTIA {
0.00349077,0.0028048,6.85969E-4
      }
      @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
      @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
      @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
      @DAMPING_COEFFICIENT { 1.00000e-002}
    }
    ! third crossection
    @ETA_VALUE {0.166666666666667
    }{
      @AXIAL_STIFFNESS {2874329.3
      }
      @BENDING_STIFFNESSES {
14376.483,4449.1536,  0.00000e+000
      }
      @TORSIONAL_STIFFNESS {287.79273
      }
      @SHEARING_STIFFNESSES {
148255.61,204853.89,  0.00000e+000
      }
      @MASS_PER_UNIT_SPAN {0.868099
      }
      @MOMENTS_OF_INERTIA {
0.00309244,0.00249145,6.0099E-4
      }
      @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
      @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
      @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
      @DAMPING_COEFFICIENT { 1.00000e-002}
    }
    ! fourth crossection
    @ETA_VALUE {0.25
    }{
      @AXIAL_STIFFNESS {2798176.1
      }
      @BENDING_STIFFNESSES {
12050.595,3619.274,  0.00000e+000
      }
      @TORSIONAL_STIFFNESS {230.44832
      }
      @SHEARING_STIFFNESSES {
133683.18,182702.59,  0.00000e+000
      }
```

```
      @MASS_PER_UNIT_SPAN {0.764717
      }
      @MOMENTS_OF_INERTIA {
0.00239418,0.00194005,4.54137E-4
      }
      @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
      @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
      @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
      @DAMPING_COEFFICIENT { 1.00000e-002}
    }
    ! fifth crossection
    @ETA_VALUE {0.583333333333333
     }{
      @AXIAL_STIFFNESS {2533950.1
      }
      @BENDING_STIFFNESSES {
4933.209,1200.361,  0.00000e+000
      }
      @TORSIONAL_STIFFNESS {86.631547
      }
      @SHEARING_STIFFNESSES {
88959.963,102425.33,  0.00000e+000
      }
      @MASS_PER_UNIT_SPAN {0.405624
      }
      @MOMENTS_OF_INERTIA {
6.65291E-4,5.55956E-4,1.09335E-4
      }
      @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
      @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
      @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
      @DAMPING_COEFFICIENT { 1.00000e-002}
    }
 ! sixth crossection
    @ETA_VALUE {1.0
    }{
      @AXIAL_STIFFNESS {2516997.8
      }
      @BENDING_STIFFNESSES {
3200.5721,1447.4232,  0.00000e+000
      }
      @TORSIONAL_STIFFNESS {103.30922
      }
      @SHEARING_STIFFNESSES {
95839.533,105356.08,  0.00000e+000
      }
      @MASS_PER_UNIT_SPAN {0.382589
      }
      @MOMENTS_OF_INERTIA {
4.62849E-4,3.29771E-4,1.33078E-4
      }
      @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
      @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
      @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
      @DAMPING_COEFFICIENT { 1.00000e-002}
    }
  }
}
```

E.4.3 properties.dat

```
@BEAM_PROPERTY_DEFINITION {
  @BEAM_PROPERTY_NAME {property_blade} {
    @ETA_VALUE { 0.00000e+000}{
      @AXIAL_STIFFNESS { 2.25763e+007}
      @BENDING_STIFFNESSES { 2.05222e+004,  7.16279e+005,  0.00000e+000}
      @TORSIONAL_STIFFNESS { 3.40680e+004}
      @SHEARING_STIFFNESSES { 3.51969e+006,  5.44326e+005,  0.00000e+000}
      @MASS_PER_UNIT_SPAN { 1.21391e-001}
      @MOMENTS_OF_INERTIA { 4.75000e-003,  1.24000e-004,  4.58000e-003}
```

```
  @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
  @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
  @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
  @DAMPING_COEFFICIENT { 1.00000e-002}
}
@ETA_VALUE { 6.63700e-003}{
  @AXIAL_STIFFNESS { 9.62576e+006}
  @BENDING_STIFFNESSES { 1.66765e+004,  4.58213e+005,  0.00000e+000}
  @TORSIONAL_STIFFNESS { 2.85036e+004}
  @SHEARING_STIFFNESSES { 2.57524e+006,  3.99578e+005,  0.00000e+000}
  @MASS_PER_UNIT_SPAN { 8.08940e-002}
  @MOMENTS_OF_INERTIA { 3.95300e-003,  1.09000e-004,  3.74800e-003}
  @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
  @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
  @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
  @DAMPING_COEFFICIENT { 1.00000e-002}
}
@ETA_VALUE { 2.87610e-002}{
  @AXIAL_STIFFNESS { 9.32100e+006}
  @BENDING_STIFFNESSES { 1.73069e+004,  4.59978e+005,  0.00000e+000}
  @TORSIONAL_STIFFNESS { 2.80533e+004}
  @SHEARING_STIFFNESSES { 2.59542e+006,  3.85470e+005,  0.00000e+000}
  @MASS_PER_UNIT_SPAN { 7.38580e-002}
  @MOMENTS_OF_INERTIA { 3.93600e-003,  1.02000e-004,  3.74800e-003}
  @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
  @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
  @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
  @DAMPING_COEFFICIENT { 1.00000e-002}
}
@ETA_VALUE { 5.53100e-002}{
  @AXIAL_STIFFNESS { 8.77064e+006}
  @BENDING_STIFFNESSES { 1.64016e+004,  4.81096e+005,  0.00000e+000}
  @TORSIONAL_STIFFNESS { 2.75118e+004}
  @SHEARING_STIFFNESSES { 2.57721e+006,  2.95805e+005,  0.00000e+000}
  @MASS_PER_UNIT_SPAN { 5.76700e-002}
  @MOMENTS_OF_INERTIA { 3.90500e-003,  8.60000e-005,  3.68300e-003}
  @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
  @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
  @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
  @DAMPING_COEFFICIENT { 1.00000e-002}
}
@ETA_VALUE { 8.40710e-002}{
  @AXIAL_STIFFNESS { 9.08786e+006}
  @BENDING_STIFFNESSES { 1.61721e+004,  5.42476e+005,  0.00000e+000}
  @TORSIONAL_STIFFNESS { 2.75926e+004}
  @SHEARING_STIFFNESSES { 2.59474e+006,  2.33435e+005,  0.00000e+000}
  @MASS_PER_UNIT_SPAN { 4.18910e-002}
  @MOMENTS_OF_INERTIA { 3.72900e-003,  7.30000e-005,  3.20900e-003}
  @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
  @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
  @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
  @DAMPING_COEFFICIENT { 1.00000e-002}
}
@ETA_VALUE { 9.07080e-002}{
  @AXIAL_STIFFNESS { 1.20554e+007}
  @BENDING_STIFFNESSES { 1.63290e+004,  6.70465e+005,  0.00000e+000}
  @TORSIONAL_STIFFNESS { 2.78870e+004}
  @SHEARING_STIFFNESSES { 2.56110e+006,  2.28841e+005,  0.00000e+000}
  @MASS_PER_UNIT_SPAN { 6.47260e-002}
  @MOMENTS_OF_INERTIA { 4.64500e-003,  7.50000e-005,  4.56900e-003}
  @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
  @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
  @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
  @DAMPING_COEFFICIENT { 1.00000e-002}
}
@ETA_VALUE { 1.00000e+000}{
  @AXIAL_STIFFNESS { 1.20554e+007}
  @BENDING_STIFFNESSES { 1.63290e+004,  6.70465e+005,  0.00000e+000}
  @TORSIONAL_STIFFNESS { 2.78870e+004}
  @SHEARING_STIFFNESSES { 2.56110e+006,  2.28841e+005,  0.00000e+000}
  @MASS_PER_UNIT_SPAN { 6.47260e-002}
```

```
      @MOMENTS_OF_INERTIA { 4.64500e-003,  7.50000e-005,  4.56900e-003}
      @CENTRE_OF_MASS_LOCATION { 0.00000e+000,  0.00000e+000}
      @SHEAR_CENTRE_LOCATION { 0.00000e+000,  0.00000e+000}
      @CENTROID_LOCATION { 0.00000e+000,  0.00000e+000}
      @DAMPING_COEFFICIENT { 1.00000e-002}
    }
  }
}
```

E.4.4 sensors_static_eigen.dat

```
@SENSOR_DEFINITION {
  @SENSOR_NAME {SensEigenvalue1to6} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {EIGENVALUES}
    @ETA_VALUE { 1.00000e+000}
  }
  @SENSOR_NAME {SensEigenvalue7to12} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {EIGENVALUES}
    @ETA_VALUE { 7.00000e+000}
  }
}
```

E.4.5 points.dat

```
@POINT_DEFINITION {
  @POINT_NAME {point_fuselage} {
    @COORDINATES { 0.00000e+000,  0.00000e+000, -4.00000e+000}
  }
  @POINT_NAME {point_shaft} {
    @COORDINATES { 0.00000e+000,  0.00000e+000, -2.00000e+000}
  }
  @POINT_NAME {point_hub} {
    @COORDINATES { 0.00000e+000,  0.00000e+000,  0.00000e+000}
  }
}
```

E.4.6 hub.dat

```
@BOUNDARY_CONDITION_DEFINITION {
  @BOUNDARY_CONDITION_NAME {clamp_shaft} {
    @APPLIED_TO_BODY {rvj_shaft}
    @AT_POINT {point_shaft}
    @DISPLACEMENT_BOUNDARY_CONDITIONS {     1,      1,      1}
    @ROTATION_BOUNDARY_CONDITIONS {     1,      1,      1}
    @TOPOLOGICAL_POSITION { 5.62304e-001, -3.93380e-001}
  }
}
@REVOLUTE_JOINT_DEFINITION {
  @REVOLUTE_JOINT_NAME {rvj_shaft} {
    @CONNECTED_TO_BODY {clamp_shaft}
    @AT_POINT {point_shaft}
    @CONNECTED_TO_BODY {shaft}
    @AT_POINT {point_shaft}
    @TRIAD_NAME {TRIAD_INERTIAL}
    @RELATIVE_ROTATION_NAME {rotation_shaft}
    @TOPOLOGICAL_POSITION { 5.61325e-001, -3.00745e-001}
  }
}
@RELATIVE_ROTATION_DEFINITION {
  @RELATIVE_ROTATION_NAME {rotation_shaft} {
    @CONNECTED_TO_BODY {clamp_shaft}
    @AT_POINT {point_shaft}
    @CONNECTED_TO_BODY {prd_shaft}
```

```
    @AT_POINT {point_shaft}
    @TOPOLOGICAL_POSITION { 2.32722e-001, -3.46297e-001}
  }
}
@PRESCRIBED_DISPLACEMENT_DEFINITION {
  @PRESCRIBED_DISPLACEMENT_NAME {prd_shaft} {
    @APPLIED_TO_BODY {rotation_shaft}
    @AT_POINT {point_shaft}
    @DEGREE_OF_FREEDOM {     4}
    @TIME_FUNCTION_NAME {control_rotation_shaft}
    @TOPOLOGICAL_POSITION { 2.32599e-001, -4.33545e-001}
  }
}
@RIGID_BODY_DEFINITION {
  @RIGID_BODY_NAME {shaft} {
    @CONNECTED_TO_BODY {rvj_shaft}
    @AT_POINT {point_shaft}
    @CONNECTED_TO_BODY {hub}
    @AT_POINT {point_hub}
    @TRIAD_NAME {TRIAD_INERTIAL}
    @SHAPE_NAME {shape_shaft}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamShaft}
    @ASSOCIATED_RIGID_BODY_NAME {
    shaft_for_pitchlinkQ1Q}
    @TOPOLOGICAL_POSITION { 5.61324e-001, -2.36089e-001}
  }
  @RIGID_BODY_NAME {hub} {
    @CONNECTED_TO_BODY {shaft}
    @AT_POINT {point_hub}
    @CONNECTED_TO_BODY {NULL}
    @AT_POINT {point_hub}
    @TRIAD_NAME {TRIAD_INERTIAL}
    @SHAPE_NAME {shape_hub}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamHub}
    @ASSOCIATED_RIGID_BODY_NAME {
    hub_for_tubeQ1Q,
    hub_for_flexQ1Q}
    @TOPOLOGICAL_POSITION { 5.61875e-001, -1.66536e-001}
  }
}
@SHAPE_DEFINITION {
  @SHAPE_NAME {shape_hub} {
    @SURFACE_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @SURFACE_NAME {surface_swashplate}
      @SCALING_FACTOR { 2.52000e-001,  2.52000e-001,  5.20000e-001}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
  }
  @SHAPE_NAME {shape_shaft} {
    @CURVE_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @CURVE_NAME {curve_circle}
      @SCALING_FACTOR { 0.00000e+000,  5.00000e-002,  5.00000e-002}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
  }
  @SHAPE_NAME {shape_fuselage} {
    @SURFACE_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @SURFACE_NAME {surface_fuselage}
      @SCALING_FACTOR { 5.00000e-001,  8.00000e-001,  1.00000e+000}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
  }
  @SHAPE_NAME {shape_shaft_for_pitchlink} {
    @CURVE_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @CURVE_NAME {curve_circle}
      @SCALING_FACTOR { 0.00000e+000,  5.00000e-002,  5.00000e-002}
```

```
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
  }
  @SHAPE_NAME {shape_hub_for_tube} {
    @CURVE_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @CURVE_NAME {curve_circle}
      @SCALING_FACTOR { 0.00000e+000,  2.50000e-002,  2.50000e-002}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
  }
  @SHAPE_NAME {shape_hub_for_flex} {
    @CURVE_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @CURVE_NAME {curve_circle}
      @SCALING_FACTOR { 0.00000e+000,  5.00000e-002,  5.00000e-002}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
  }
}
```

E.4.7 flexblade.dat

```
@ASSOCIATED_RIGID_BODY_DEFINITION {
  @ASSOCIATED_RIGID_BODY_NAME {hub_for_flexQ1Q} {
    @CONNECTED_TO_BODY {shaft}
    @AT_POINT {point_hub}
    @CONNECTED_TO_BODY {flexbeamQ1Q}
    @AT_POINT {point_flex_rootQ1Q}
    @SHAPE_NAME {shape_hub_for_flex}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamArbHub2Flexbeam}
    @TOPOLOGICAL_POSITION { 1.03160e+000, -3.29391e-001}
  }
}
@BEAM_DEFINITION {
  @BEAM_NAME {flexbeamQ1Q} {
    @CONNECTED_TO_BODY {hub_for_flexQ1Q}
    @AT_POINT {point_flex_rootQ1Q}
    @CONNECTED_TO_BODY {flex_blade_connectorQ1Q}
    @AT_POINT {point_blade_rootQ1Q}
    @CURVE_NAME {curve_flexbeamQ1Q}
    @BEAM_PROPERTY_NAME {property_flexbeam}
    @SHAPE_NAME {shape_flexbeam}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamFlexBeam}
    @TOPOLOGICAL_POSITION { 1.39063e+000, -3.27584e-001}
  }
}
@RIGID_BODY_DEFINITION {
  @RIGID_BODY_NAME {flex_blade_connectorQ1Q} {
    @CONNECTED_TO_BODY {unj_tubeQ1Q}
    @AT_POINT {point_blade_rootQ1Q}
    @CONNECTED_TO_BODY {flexbeamQ1Q}
    @AT_POINT {point_blade_rootQ1Q}
    @TRIAD_NAME {triad_flex_blade_connectorQ1Q}
    @ASSOCIATED_RIGID_BODY_NAME {
    connector_for_bladeQ1Q}
    @TOPOLOGICAL_POSITION { 1.39086e+000, -2.19021e-001}
  }
}
@ASSOCIATED_RIGID_BODY_DEFINITION {
  @ASSOCIATED_RIGID_BODY_NAME {connector_for_bladeQ1Q} {
    @CONNECTED_TO_BODY {unj_tubeQ1Q}
    @AT_POINT {point_blade_rootQ1Q}
    @CONNECTED_TO_BODY {bladeQ1Q}
    @AT_POINT {point_blade_rootQ1Q}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamArbTube2Blade}
    @TOPOLOGICAL_POSITION { 1.71930e+000, -3.81304e-002}
  }
```

```
}
@BEAM_DEFINITION {
  @BEAM_NAME {bladeQ1Q} {
    @CONNECTED_TO_BODY {connector_for_bladeQ1Q}
    @AT_POINT {point_blade_rootQ1Q}
    @CONNECTED_TO_BODY {tipwtQ1Q}
    @AT_POINT {point_blade_tipQ1Q}
    @CURVE_NAME {curve_bladeQ1Q}
    @BEAM_PROPERTY_NAME {property_blade}
    @SHAPE_NAME {shape_blade}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamBlade}
    @TOPOLOGICAL_POSITION { 1.71960e+000,  4.60581e-002}
  }
}
@RIGID_BODY_DEFINITION {
  @RIGID_BODY_NAME {tipwtQ1Q} {
    @CONNECTED_TO_BODY {bladeQ1Q}
    @AT_POINT {point_blade_tipQ1Q}
    @CONNECTED_TO_BODY {NULL}
    @AT_POINT {point_blade_tipQ1Q}
    @TRIAD_NAME {triad_tipwt}
    @MASS_PROPERTY_NAME {property_tipwt}
    @SHAPE_NAME {shape_tipwt}
    @TOPOLOGICAL_POSITION { 1.71960e+000,  1.10142e-001}
  }
}
@TRIAD_DEFINITION {
  @TRIAD_NAME {triad_flex_blade_connectorQ1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @IS_DEFINED_IN_FRAME {FrameBladePreconeQ1Q}
  }
  @TRIAD_NAME {triad_tipwt} {
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @IS_DEFINED_IN_FRAME {FrameBladeSweepQ1Q}
  }
}
@CURVE_DEFINITION {
  @CURVE_NAME {curve_flexbeamQ1Q} {
    @IS_DEFINED_IN_FRAME {FrameBladePreconeQ1Q}
    @POINT_DEFINITION {
      @NUMBER_OF_CONTROL_POINTS {     2}
      @DEGREE_OF_CURVE {     1}
      @RATIONAL_CURVE_FLAG {NO}
      @END_POINT_0 {point_flex_rootQ1Q}
      @END_POINT_1 {point_blade_rootQ1Q}
    }
    @TRIAD_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
      @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    }
    @CURVE_MESH_PARAMETERS_NAME {curve_flexbeam_mesh_parameters}
  }
  @CURVE_NAME {curve_bladeQ1Q} {
    @IS_DEFINED_IN_FRAME {FrameBladeSweepQ1Q}
    @POINT_DEFINITION {
      @NUMBER_OF_CONTROL_POINTS {     2}
      @DEGREE_OF_CURVE {     1}
      @RATIONAL_CURVE_FLAG {NO}
      @END_POINT_0 {point_blade_rootQ1Q}
      @END_POINT_1 {point_blade_tipQ1Q}
    }
    @TRIAD_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
      @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
```

```
    }
    @CURVE_MESH_PARAMETERS_NAME {curve_blade_mesh_parameters}
  }
}
@CURVE_MESH_PARAMETERS_DEFINITION {
  @CURVE_MESH_PARAMETERS_NAME {curve_flexbeam_mesh_parameters} {
    @NUMBER_OF_ELEMENTS {     6}
    @ORDER_OF_ELEMENTS {     3}
  }
  @CURVE_MESH_PARAMETERS_NAME {curve_blade_mesh_parameters} {
    @NUMBER_OF_ELEMENTS {     3}
    @ORDER_OF_ELEMENTS {     3}
  }
}
@SHAPE_DEFINITION {
  @SHAPE_NAME {shape_flexbeam} {
    @CURVE_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @CURVE_NAME {hanson_flex}
      @SCALING_FACTOR { 0.00000e+000,  1.80000e-001,  1.80000e-001}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
      @ETA_VALUE { 1.00000e+000}
      @CURVE_NAME {hanson_flex}
      @SCALING_FACTOR { 0.00000e+000,  1.50000e-001,  1.50000e-001}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
  }
  @SHAPE_NAME {shape_blade} {
    @CURVE_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @CURVE_NAME {curve_naca0010}
      @SCALING_FACTOR { 0.00000e+000,  5.00000e-001,  1.75000e+000}
      @ORIGIN { 0.00000e+000, -6.50000e-001,  0.00000e+000}
      @ETA_VALUE { 1.00000e+000}
      @CURVE_NAME {curve_naca0010}
      @SCALING_FACTOR { 0.00000e+000,  5.00000e-001,  1.75000e+000}
      @ORIGIN { 0.00000e+000, -6.50000e-001,  0.00000e+000}
    }
  }
  @SHAPE_NAME {shape_tipwt} {
    @SURFACE_DEFINITION {
      @ETA_VALUE { 0.00000e+000}
      @SURFACE_NAME {surface_nacelle}
      @SCALING_FACTOR { 2.00000e-001,  2.00000e-001,  2.00000e-002}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
  }
}
```

E.4.8 torquetube.dat

```
@ASSOCIATED_RIGID_BODY_DEFINITION {
  @ASSOCIATED_RIGID_BODY_NAME {hub_for_tubeQ1Q} {
    @CONNECTED_TO_BODY {shaft}
    @AT_POINT {point_hub}
    @CONNECTED_TO_BODY {prj_tubeQ1Q}
    @AT_POINT {point_flex_rootQ1Q}
    @SHAPE_NAME {shape_hub_for_tube}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamArbHub2Tube}
    @TOPOLOGICAL_POSITION { 1.02756e+000, -2.01313e-001}
  }
}
@PRISMATIC_JOINT_DEFINITION {
  @PRISMATIC_JOINT_NAME {prj_tubeQ1Q} {
    @CONNECTED_TO_BODY {hub_for_tubeQ1Q}
    @AT_POINT {point_flex_rootQ1Q}
    @CONNECTED_TO_BODY {spj_tubeQ1Q}
    @AT_POINT {point_flex_rootQ1Q}
```

```
    @TRIAD_NAME {triad_prj_tubeQ1Q}
    @TOPOLOGICAL_POSITION { 1.02444e+000, -9.65541e-002}
  }
}
@SPHERICAL_JOINT_DEFINITION {
  @SPHERICAL_JOINT_NAME {spj_tubeQ1Q} {
    @CONNECTED_TO_BODY {prj_tubeQ1Q}
    @AT_POINT {point_flex_rootQ1Q}
    @CONNECTED_TO_BODY {tube_assemblyQ1Q}
    @AT_POINT {point_flex_rootQ1Q}
    @TRIAD_NAME {triad_spj_tubeQ1Q}
    @TOPOLOGICAL_POSITION { 1.02499e+000, -1.26740e-002}
  }
}
@RIGID_BODY_DEFINITION {
  @RIGID_BODY_NAME {tube_assemblyQ1Q} {
    @CONNECTED_TO_BODY {spj_tubeQ1Q}
    @AT_POINT {point_flex_rootQ1Q}
    @CONNECTED_TO_BODY {unj_tubeQ1Q}
    @AT_POINT {point_blade_rootQ1Q}
    @TRIAD_NAME {triad_tube_assemblyQ1Q}
    @SHAPE_NAME {shape_tube}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamTube}
    @ASSOCIATED_RIGID_BODY_NAME {
    tube_assembly_for_pitchlinkQ1Q}
    @TOPOLOGICAL_POSITION { 1.02499e+000,  1.44739e-001}
  }
}
@UNIVERSAL_JOINT_DEFINITION {
  @UNIVERSAL_JOINT_NAME {unj_tubeQ1Q} {
    @CONNECTED_TO_BODY {tube_assemblyQ1Q}
    @AT_POINT {point_blade_rootQ1Q}
    @CONNECTED_TO_BODY {flex_blade_connectorQ1Q}
    @AT_POINT {point_blade_rootQ1Q}
    @TRIAD_NAME {triad_unj_tube_lQ1Q, triad_unj_tube_rQ1Q}
    @TOPOLOGICAL_POSITION { 1.39029e+000,  1.44862e-001}
  }
}
@TRIAD_DEFINITION {
  @TRIAD_NAME {triad_prj_tubeQ1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 1.00000e+000,  0.00000e+000,  0.00000e+000}
    @IS_DEFINED_IN_FRAME {FrameBladePreconeQ1Q}
  }
  @TRIAD_NAME {triad_spj_tubeQ1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @IS_DEFINED_IN_FRAME {FrameBladePreconeQ1Q}
  }
  @TRIAD_NAME {triad_tube_assemblyQ1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 1.00000e+000,  0.00000e+000,  0.00000e+000}
    @IS_DEFINED_IN_FRAME {FrameBladePreconeQ1Q}
  }
  @TRIAD_NAME {triad_unj_tube_lQ1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @IS_DEFINED_IN_FRAME {FrameBladePreconeQ1Q}
  }
  @TRIAD_NAME {triad_unj_tube_rQ1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @IS_DEFINED_IN_FRAME {FrameBladePreconeQ1Q}
  }
}
@SHAPE_DEFINITION {
  @SHAPE_NAME {shape_tube} {
    @CURVE_DEFINITION {
```

```
      @ETA_VALUE { 0.00000e+000}
      @CURVE_NAME {curve_circle}
      @SCALING_FACTOR { 0.00000e+000,  5.00000e-002,  5.00000e-002}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
      @ETA_VALUE { 1.00000e+000}
      @CURVE_NAME {curve_circle}
      @SCALING_FACTOR { 0.00000e+000,  5.00000e-002,  5.00000e-002}
      @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
  }
}
```

E.4.9 pitchlink.dat

```
@ASSOCIATED_RIGID_BODY_DEFINITION {
  @ASSOCIATED_RIGID_BODY_NAME {shaft_for_pitchlinkQ1Q} {
    @CONNECTED_TO_BODY {rvj_shaft}
    @AT_POINT {point_shaft}
    @CONNECTED_TO_BODY {unj_pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @SHAPE_NAME {shape_shaft_for_pitchlink}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamArbShaft2Pitchlink}
    @TOPOLOGICAL_POSITION {-9.85137e-002, -2.68340e-001}
  }
}
@UNIVERSAL_JOINT_DEFINITION {
  @UNIVERSAL_JOINT_NAME {unj_pitchlinkQ1Q} {
    @CONNECTED_TO_BODY {shaft_for_pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @CONNECTED_TO_BODY {prj_pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @TRIAD_NAME {triad_unj_pitchlink_0Q1Q, triad_unj_pitchlink_1Q1Q}
    @TOPOLOGICAL_POSITION {-9.85134e-002,  1.29552e-001}
  }
}
@PRISMATIC_JOINT_DEFINITION {
  @PRISMATIC_JOINT_NAME {prj_pitchlinkQ1Q} {
    @CONNECTED_TO_BODY {prj_for_pitchQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @CONNECTED_TO_BODY {unj_pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @TRIAD_NAME {triad_prj_pitchlinkQ1Q}
    @RELATIVE_DISPLACEMENT_NAME {PrjRelativeMotionSQ1Q}
    @TOPOLOGICAL_POSITION {-9.85136e-002,  4.05075e-001}
  }
  @PRISMATIC_JOINT_NAME {prj_for_pitchQ1Q} {
    @CONNECTED_TO_BODY {prj_pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @CONNECTED_TO_BODY {pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @TRIAD_NAME {triad_prj_pitchlinkQ1Q}
    @RELATIVE_DISPLACEMENT_NAME {ControlPitchQ1Q}
    @TOPOLOGICAL_POSITION { 6.91492e-001,  4.04523e-001}
  }
}
@RIGID_BODY_DEFINITION {
  @RIGID_BODY_NAME {pitchlinkQ1Q} {
    @CONNECTED_TO_BODY {spj_pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_tubeQ1Q}
    @CONNECTED_TO_BODY {prj_for_pitchQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @TRIAD_NAME {triad_pitchlinkQ1Q}
    @SHAPE_NAME {shape_shaft}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamPitchlink}
    @TOPOLOGICAL_POSITION { 6.89839e-001,  2.96398e-001}
  }
```

```
}
@SPHERICAL_JOINT_DEFINITION {
  @SPHERICAL_JOINT_NAME {spj_pitchlinkQ1Q} {
    @CONNECTED_TO_BODY {pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_tubeQ1Q}
    @CONNECTED_TO_BODY {tube_assembly_for_pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_tubeQ1Q}
    @TRIAD_NAME {triad_spj_pitchlinkQ1Q}
    @TOPOLOGICAL_POSITION { 6.90390e-001,  1.84353e-001}
  }
}
@ASSOCIATED_RIGID_BODY_DEFINITION {
  @ASSOCIATED_RIGID_BODY_NAME {tube_assembly_for_pitchlinkQ1Q} {
    @CONNECTED_TO_BODY {spj_tubeQ1Q}
    @AT_POINT {point_flex_rootQ1Q}
    @CONNECTED_TO_BODY {spj_pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_tubeQ1Q}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamArbTube2Pitchlink}
    @TOPOLOGICAL_POSITION { 6.90820e-001,  6.70428e-002}
  }
}
@RELATIVE_DISPLACEMENT_DEFINITION {
  @RELATIVE_DISPLACEMENT_NAME {PrjRelativeMotionSQ1Q} {
    @CONNECTED_TO_BODY {prj_for_pitchQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @CONNECTED_TO_BODY {NULL}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @SPRING_NAME {spring_pitchlink}
    @TOPOLOGICAL_POSITION { 2.97255e-001,  4.91343e-001}
  }
  @RELATIVE_DISPLACEMENT_NAME {ControlPitchQ1Q} {
    @CONNECTED_TO_BODY {prj_pitchlinkQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @CONNECTED_TO_BODY {PrdPitchInputQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @TOPOLOGICAL_POSITION { 2.97255e-001,  3.03622e-001}
  }
}
@PRESCRIBED_DISPLACEMENT_DEFINITION {
  @PRESCRIBED_DISPLACEMENT_NAME {PrdPitchInputQ1Q} {
    @APPLIED_TO_BODY {ControlPitchQ1Q}
    @AT_POINT {point_pitchlink_shaftQ1Q}
    @DEGREE_OF_FREEDOM {     1}
    @TIME_FUNCTION_NAME {prj_control_input}
    @TOPOLOGICAL_POSITION { 2.97255e-001,  1.63536e-001}
  }
}
@SPRING_DEFINITION {
  @SPRING_NAME {spring_pitchlink} {
    @SPRING_TYPE {LINEAR}
    @SPRING_DEFINITION_TYPE {CHEBYCHEV}
    @APPROXIMATION_RANGE {-1.00000e+000,  1.00000e+000}
    @CHEBYCHEV_COEFFICIENTS {
     0.00000e+000,  0.00000e+000}
  }
}
@TRIAD_DEFINITION {
  @TRIAD_NAME {triad_pitchlinkQ1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
  @TRIAD_NAME {triad_spj_pitchlinkQ1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
  @TRIAD_NAME {triad_prj_pitchlinkQ1Q} {
```

```
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
  @TRIAD_NAME {triad_unj_pitchlink_0Q1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
  @TRIAD_NAME {triad_unj_pitchlink_1Q1Q} {
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 1.00000e+000,  0.00000e+000,  0.00000e+000}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
}
```

E.4.10 shapes.dat

```
@CURVE_DEFINITION {
  @CURVE_NAME {curve_circle} {
    @IS_DEFINED_IN_FRAME {INERTIAL}
    @POINT_DEFINITION {
      @NUMBER_OF_CONTROL_POINTS {    10}
      @DEGREE_OF_CURVE {     3}
      @RATIONAL_CURVE_FLAG {YES}
      @COORDINATES { 0.00000e+000,  0.00000e+000, -1.00000e+000,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -8.66025e-001, -1.00000e+000,  6.66667e-001}
      @COORDINATES { 0.00000e+000, -1.29904e+000, -2.50000e-001,  6.66667e-001}
      @COORDINATES { 0.00000e+000, -8.66025e-001,  5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -4.33013e-001,  1.25000e+000,  6.66667e-001}
      @COORDINATES { 0.00000e+000,  4.33013e-001,  1.25000e+000,  6.66667e-001}
      @COORDINATES { 0.00000e+000,  8.66025e-001,  5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000,  1.29904e+000, -2.50000e-001,  6.66667e-001}
      @COORDINATES { 0.00000e+000,  8.66025e-001, -1.00000e+000,  6.66667e-001}
      @COORDINATES { 0.00000e+000,  0.00000e+000, -1.00000e+000,  1.00000e+000}
    }
    @KNOT_SEQUENCE_DEFINITION {
      @KNOT_SEQUENCE {
       0.00000e+000,  0.00000e+000,  0.00000e+000,  0.00000e+000,  3.33333e-001,
        3.33333e-001,
       3.33333e-001,  6.66667e-001,  6.66667e-001,  6.66667e-001,  1.00000e+000,
        1.00000e+000,
       1.00000e+000,  1.00000e+000}
    }
  }
  @CURVE_NAME {hanson_flex} {
    @IS_DEFINED_IN_FRAME {INERTIAL}
    @POINT_DEFINITION {
      @NUMBER_OF_CONTROL_POINTS {    25}
      @DEGREE_OF_CURVE {     3}
      @RATIONAL_CURVE_FLAG {YES}
      @COORDINATES { 0.00000e+000,  2.50000e-001,  5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000,  1.25000e-001,  5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -1.25000e-001,  5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -2.50000e-001,  5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -2.50000e-001,  3.53553e-001,  8.04738e-001}
      @COORDINATES { 0.00000e+000, -3.53553e-001,  2.50000e-001,  8.04738e-001}
      @COORDINATES { 0.00000e+000, -5.00000e-001,  2.50000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -5.00000e-001,  1.25000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -5.00000e-001, -1.25000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -5.00000e-001, -2.50000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -3.53553e-001, -2.50000e-001,  8.04738e-001}
      @COORDINATES { 0.00000e+000, -2.50000e-001, -3.53553e-001,  8.04738e-001}
      @COORDINATES { 0.00000e+000, -2.50000e-001, -5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -1.25000e-001, -5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000,  1.25000e-001, -5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000,  2.50000e-001, -5.00000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000,  2.50000e-001, -3.53553e-001,  8.04738e-001}
      @COORDINATES { 0.00000e+000,  3.53553e-001, -2.50000e-001,  8.04738e-001}
```

```
      @COORDINATES { 0.00000e+000,  5.00000e-001, -2.50000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000,  5.00000e-001, -1.25000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000,  5.00000e-001,  1.25000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000,  5.00000e-001,  2.50000e-001,  1.00000e+000}
      @COORDINATES { 0.00000e+000,  3.53553e-001,  2.50000e-001,  8.04738e-001}
      @COORDINATES { 0.00000e+000,  2.50000e-001,  3.53553e-001,  8.04738e-001}
      @COORDINATES { 0.00000e+000,  2.50000e-001,  5.00000e-001,  1.00000e+000}
    }
    @KNOT_SEQUENCE_DEFINITION {
      @KNOT_SEQUENCE {
       0.00000e+000,  0.00000e+000,  0.00000e+000,  0.00000e+000,  1.25000e-001,
       1.25000e-001,
       1.25000e-001,  2.50000e-001,  2.50000e-001,  2.50000e-001,  3.75000e-001,
        3.75000e-001,
       3.75000e-001,  5.00000e-001,  5.00000e-001,  5.00000e-001,  6.25000e-001,
        6.25000e-001,
       6.25000e-001,  7.50000e-001,  7.50000e-001,  7.50000e-001,  8.75000e-001,
        8.75000e-001,
       8.75000e-001,  1.00000e+000,  1.00000e+000,  1.00000e+000,  1.00000e+000}
    }
  }
  @CURVE_NAME {curve_naca0010} {
    @IS_DEFINED_IN_FRAME {INERTIAL}
    @POINT_DEFINITION {
      @NUMBER_OF_CONTROL_POINTS {     5}
      @DEGREE_OF_CURVE {     3}
      @RATIONAL_CURVE_FLAG {NO}
      @COORDINATES { 0.00000e+000,  0.00000e+000,  0.00000e+000}
      @COORDINATES { 0.00000e+000,  7.59500e-001,  8.70000e-002}
      @COORDINATES { 0.00000e+000,  1.24050e+000,  0.00000e+000}
      @COORDINATES { 0.00000e+000,  7.59500e-001, -8.70000e-002}
      @COORDINATES { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    }
    @KNOT_SEQUENCE_DEFINITION {
      @KNOT_SEQUENCE {
       0.00000e+000,  0.00000e+000,  0.00000e+000,  0.00000e+000,  5.00000e-001,
       1.00000e+000,
       1.00000e+000,  1.00000e+000,  1.00000e+000}
    }
  }
}
@SURFACE_DEFINITION {
  @SURFACE_NAME {surface_fuselage} {
    @IS_DEFINED_IN_FRAME {INERTIAL}
    @POINT_DEFINITION {
      @NUMBER_OF_CONTROL_POINTS {     9,      5}
      @DEGREE_OF_CURVE {     2,      2}
      @RATIONAL_CURVE_FLAG {YES}
      @COORDINATES {-5.99801e+000, -1.65560e-002,  0.00000e+000,  1.00000e+000}
      @COORDINATES {-5.99802e+000, -1.65561e-002,  0.00000e+000,  7.07107e-001}
      @COORDINATES {-5.99801e+000, -1.65560e-002,  0.00000e+000,  1.00000e+000}
      @COORDINATES {-5.99802e+000, -1.65561e-002,  0.00000e+000,  7.07107e-001}
      @COORDINATES {-5.99801e+000, -1.65560e-002,  0.00000e+000,  1.00000e+000}
      @COORDINATES {-5.99802e+000, -1.65561e-002,  0.00000e+000,  7.07107e-001}
      @COORDINATES {-5.99801e+000, -1.65560e-002,  0.00000e+000,  1.00000e+000}
      @COORDINATES {-5.99802e+000, -1.65561e-002,  0.00000e+000,  7.07107e-001}
      @COORDINATES {-5.99801e+000, -1.65560e-002,  0.00000e+000,  1.00000e+000}
      @COORDINATES {-5.99802e+000, -1.65561e-002,  1.00000e+000,  7.07107e-001}
      @COORDINATES {-5.99802e+000, -1.01656e+000,  1.00000e+000,  5.00000e-001}
      @COORDINATES {-5.99802e+000, -1.01656e+000,  2.77556e-016,  7.07107e-001}
      @COORDINATES {-5.99802e+000, -1.01656e+000, -1.00000e+000,  5.00000e-001}
      @COORDINATES {-5.99802e+000, -1.65561e-002, -1.00000e+000,  7.07107e-001}
      @COORDINATES {-5.99802e+000,  9.83444e-001, -1.00000e+000,  5.00000e-001}
      @COORDINATES {-5.99802e+000,  9.83444e-001,  1.11022e-016,  7.07107e-001}
      @COORDINATES {-5.99802e+000,  9.83444e-001,  1.00000e+000,  5.00000e-001}
      @COORDINATES {-5.99802e+000, -1.65561e-002,  1.00000e+000,  7.07107e-001}
      @COORDINATES { 0.00000e+000,  0.00000e+000,  1.00000e+000,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -1.00000e+000,  1.00000e+000,  7.07107e-001}
      @COORDINATES { 0.00000e+000, -1.00000e+000,  2.77556e-016,  1.00000e+000}
      @COORDINATES { 0.00000e+000, -1.00000e+000, -1.00000e+000,  7.07107e-001}
      @COORDINATES { 0.00000e+000, -1.66533e-016, -1.00000e+000,  1.00000e+000}
```

```
    @COORDINATES { 0.00000e+000,  1.00000e+000, -1.00000e+000,  7.07107e-001}
    @COORDINATES { 0.00000e+000,  1.00000e+000,  1.11022e-016,  1.00000e+000}
    @COORDINATES { 0.00000e+000,  1.00000e+000,  1.00000e+000,  7.07107e-001}
    @COORDINATES { 0.00000e+000,  0.00000e+000,  1.00000e+000,  1.00000e+000}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  1.00000e+000,  7.07107e-001}
    @COORDINATES { 4.00000e+000, -1.00000e+000,  1.00000e+000,  5.00000e-001}
    @COORDINATES { 4.00000e+000, -1.00000e+000,  2.77556e-016,  7.07107e-001}
    @COORDINATES { 4.00000e+000, -1.00000e+000, -1.00000e+000,  5.00000e-001}
    @COORDINATES { 4.00000e+000, -1.66533e-016, -1.00000e+000,  7.07107e-001}
    @COORDINATES { 4.00000e+000,  1.00000e+000, -1.00000e+000,  5.00000e-001}
    @COORDINATES { 4.00000e+000,  1.00000e+000,  1.11022e-016,  7.07107e-001}
    @COORDINATES { 4.00000e+000,  1.00000e+000,  1.00000e+000,  5.00000e-001}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  1.00000e+000,  7.07107e-001}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES { 4.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
  }
  @KNOT_SEQUENCE_DEFINITION {
    @KNOT_SEQUENCE {
     0.00000e+000,  0.00000e+000,  0.00000e+000,  2.50000e-001,  2.50000e-001,
      5.00000e-001,
     5.00000e-001,  7.50000e-001,  7.50000e-001,  1.00000e+000,  1.00000e+000,
     1.00000e+000}
    @KNOT_SEQUENCE {
     0.00000e+000,  0.00000e+000,  0.00000e+000,  5.00000e-001,  5.00000e-001,
      1.00000e+000,
     1.00000e+000,  1.00000e+000}
  }
}
@SURFACE_NAME {surface_blade_edge} {
  @IS_DEFINED_IN_FRAME {INERTIAL}
  @POINT_DEFINITION {
    @NUMBER_OF_CONTROL_POINTS {    13,     13}
    @DEGREE_OF_CURVE {     3,      3}
    @RATIONAL_CURVE_FLAG {NO}
    @COORDINATES {-4.86002e-002, -5.00000e-001,  7.55729e-002}
    @COORDINATES { 4.10004e-002, -5.00000e-001,  7.40001e-002}
    @COORDINATES { 1.30800e-001, -5.00000e-001,  7.24253e-002}
    @COORDINATES { 2.21000e-001, -5.00000e-001,  7.08178e-002}
    @COORDINATES { 3.12100e-001, -5.00000e-001,  6.91348e-002}
    @COORDINATES { 4.04300e-001, -5.00000e-001,  6.73433e-002}
    @COORDINATES { 4.97600e-001, -5.00000e-001,  6.54691e-002}
    @COORDINATES { 5.92100e-001, -5.00000e-001,  6.36629e-002}
    @COORDINATES { 6.87800e-001, -5.00000e-001,  6.21679e-002}
    @COORDINATES { 7.83800e-001, -5.00000e-001,  6.10359e-002}
    @COORDINATES { 8.79400e-001, -5.00000e-001,  6.01240e-002}
    @COORDINATES { 9.73900e-001, -5.00000e-001,  5.92269e-002}
    @COORDINATES { 1.06810e+000, -5.00000e-001,  5.83273e-002}
    @COORDINATES {-4.82998e-002, -5.00000e-001,  6.62309e-002}
    @COORDINATES { 4.11997e-002, -5.00000e-001,  6.46685e-002}
    @COORDINATES { 1.31000e-001, -5.00000e-001,  6.31208e-002}
    @COORDINATES { 2.21200e-001, -5.00000e-001,  6.15596e-002}
    @COORDINATES { 3.12200e-001, -5.00000e-001,  5.99511e-002}
    @COORDINATES { 4.04100e-001, -5.00000e-001,  5.82641e-002}
    @COORDINATES { 4.97100e-001, -5.00000e-001,  5.65039e-002}
    @COORDINATES { 5.91100e-001, -5.00000e-001,  5.47561e-002}
    @COORDINATES { 6.86000e-001, -5.00000e-001,  5.32285e-002}
    @COORDINATES { 7.81700e-001, -5.00000e-001,  5.21447e-002}
    @COORDINATES { 8.76700e-001, -5.00000e-001,  5.11739e-002}
    @COORDINATES { 9.70700e-001, -5.00000e-001,  5.01882e-002}
    @COORDINATES { 1.06440e+000, -5.00000e-001,  4.92345e-002}
    @COORDINATES {-4.80003e-002, -5.00000e-001,  5.68566e-002}
    @COORDINATES { 4.14000e-002, -5.00000e-001,  5.53075e-002}
    @COORDINATES { 1.30900e-001, -5.00000e-001,  5.37984e-002}
```

```
@COORDINATES { 2.21000e-001, -5.00000e-001,  5.22994e-002}
@COORDINATES { 3.11700e-001, -5.00000e-001,  5.07792e-002}
@COORDINATES { 4.03400e-001, -5.00000e-001,  4.91924e-002}
@COORDINATES { 4.96100e-001, -5.00000e-001,  4.75089e-002}
@COORDINATES { 5.89700e-001, -5.00000e-001,  4.57920e-002}
@COORDINATES { 6.84300e-001, -5.00000e-001,  4.42601e-002}
@COORDINATES { 7.79600e-001, -5.00000e-001,  4.31881e-002}
@COORDINATES { 8.74100e-001, -5.00000e-001,  4.21547e-002}
@COORDINATES { 9.67300e-001, -5.00000e-001,  4.10627e-002}
@COORDINATES { 1.06040e+000, -5.00000e-001,  4.00386e-002}
@COORDINATES {-4.78001e-002, -5.00000e-001,  4.73916e-002}
@COORDINATES { 4.14000e-002, -5.00000e-001,  4.58633e-002}
@COORDINATES { 1.30700e-001, -5.00000e-001,  4.43956e-002}
@COORDINATES { 2.20500e-001, -5.00000e-001,  4.29544e-002}
@COORDINATES { 3.11000e-001, -5.00000e-001,  4.15137e-002}
@COORDINATES { 4.02400e-001, -5.00000e-001,  4.00509e-002}
@COORDINATES { 4.94800e-001, -5.00000e-001,  3.84631e-002}
@COORDINATES { 5.88200e-001, -5.00000e-001,  3.67691e-002}
@COORDINATES { 6.82300e-001, -5.00000e-001,  3.52663e-002}
@COORDINATES { 7.77000e-001, -5.00000e-001,  3.40649e-002}
@COORDINATES { 8.71000e-001, -5.00000e-001,  3.29472e-002}
@COORDINATES { 9.63100e-001, -5.00000e-001,  3.16831e-002}
@COORDINATES { 1.05510e+000, -5.00000e-001,  3.05485e-002}
@COORDINATES {-4.74005e-002, -5.00000e-001,  3.77937e-002}
@COORDINATES { 4.14000e-002, -5.00000e-001,  3.62989e-002}
@COORDINATES { 1.30400e-001, -5.00000e-001,  3.48687e-002}
@COORDINATES { 2.19900e-001, -5.00000e-001,  3.34708e-002}
@COORDINATES { 3.10200e-001, -5.00000e-001,  3.20824e-002}
@COORDINATES { 4.01400e-001, -5.00000e-001,  3.06753e-002}
@COORDINATES { 4.93600e-001, -5.00000e-001,  2.91544e-002}
@COORDINATES { 5.86600e-001, -5.00000e-001,  2.75261e-002}
@COORDINATES { 6.80100e-001, -5.00000e-001,  2.60017e-002}
@COORDINATES { 7.73400e-001, -5.00000e-001,  2.46062e-002}
@COORDINATES { 8.66000e-001, -5.00000e-001,  2.32483e-002}
@COORDINATES { 9.57900e-001, -5.00000e-001,  2.19637e-002}
@COORDINATES { 1.04910e+000, -5.00000e-001,  2.07983e-002}
@COORDINATES {-4.71001e-002, -5.00000e-001,  2.80571e-002}
@COORDINATES { 4.12998e-002, -5.00000e-001,  2.66094e-002}
@COORDINATES { 1.30000e-001, -5.00000e-001,  2.52118e-002}
@COORDINATES { 2.19200e-001, -5.00000e-001,  2.38401e-002}
@COORDINATES { 3.09400e-001, -5.00000e-001,  2.24741e-002}
@COORDINATES { 4.00600e-001, -5.00000e-001,  2.10719e-002}
@COORDINATES { 4.92800e-001, -5.00000e-001,  1.95897e-002}
@COORDINATES { 5.85500e-001, -5.00000e-001,  1.80443e-002}
@COORDINATES { 6.78300e-001, -5.00000e-001,  1.65034e-002}
@COORDINATES { 7.70500e-001, -5.00000e-001,  1.50027e-002}
@COORDINATES { 8.62000e-001, -5.00000e-001,  1.35304e-002}
@COORDINATES { 9.53800e-001, -5.00000e-001,  1.22034e-002}
@COORDINATES { 1.04460e+000, -5.00000e-001,  1.09440e-002}
@COORDINATES {-4.72002e-002, -5.00000e-001,  1.82151e-002}
@COORDINATES { 4.09002e-002, -5.00000e-001,  1.68231e-002}
@COORDINATES { 1.29300e-001, -5.00000e-001,  1.54558e-002}
@COORDINATES { 2.18300e-001, -5.00000e-001,  1.41013e-002}
@COORDINATES { 3.08400e-001, -5.00000e-001,  1.27463e-002}
@COORDINATES { 3.99900e-001, -5.00000e-001,  1.13664e-002}
@COORDINATES { 4.92400e-001, -5.00000e-001,  9.93023e-003}
@COORDINATES { 5.85300e-001, -5.00000e-001,  8.42881e-003}
@COORDINATES { 6.78100e-001, -5.00000e-001,  6.89453e-003}
@COORDINATES { 7.70200e-001, -5.00000e-001,  5.38431e-003}
@COORDINATES { 8.61400e-001, -5.00000e-001,  3.94169e-003}
@COORDINATES { 9.53200e-001, -5.00000e-001,  2.51594e-003}
@COORDINATES { 1.04410e+000, -5.00000e-001,  1.09842e-003}
@COORDINATES {-4.82998e-002, -5.00000e-001,  8.33097e-003}
@COORDINATES { 3.97997e-002, -5.00000e-001,  6.99142e-003}
@COORDINATES { 1.28100e-001, -5.00000e-001,  5.65452e-003}
@COORDINATES { 2.16600e-001, -5.00000e-001,  4.31895e-003}
@COORDINATES { 3.07000e-001, -5.00000e-001,  2.97234e-003}
@COORDINATES { 3.99200e-001, -5.00000e-001,  1.62257e-003}
@COORDINATES { 4.92500e-001, -5.00000e-001,  2.38767e-004}
```

```
@COORDINATES { 5.86100e-001, -5.00000e-001, -1.22897e-003}
@COORDINATES { 6.79700e-001, -5.00000e-001, -2.76427e-003}
@COORDINATES { 7.72800e-001, -5.00000e-001, -4.29878e-003}
@COORDINATES { 8.64900e-001, -5.00000e-001, -5.79826e-003}
@COORDINATES { 9.56800e-001, -5.00000e-001, -7.28863e-003}
@COORDINATES { 1.04820e+000, -5.00000e-001, -8.77034e-003}
@COORDINATES {-5.09996e-002, -5.00000e-001, -1.52807e-003}
@COORDINATES { 3.74002e-002, -5.00000e-001, -2.81923e-003}
@COORDINATES { 1.25500e-001, -5.00000e-001, -4.12638e-003}
@COORDINATES { 2.13800e-001, -5.00000e-001, -5.44097e-003}
@COORDINATES { 3.05000e-001, -5.00000e-001, -6.79310e-003}
@COORDINATES { 3.98700e-001, -5.00000e-001, -8.10939e-003}
@COORDINATES { 4.93100e-001, -5.00000e-001, -9.41399e-003}
@COORDINATES { 5.87300e-001, -5.00000e-001, -1.08618e-002}
@COORDINATES { 6.81900e-001, -5.00000e-001, -1.24642e-002}
@COORDINATES { 7.76900e-001, -5.00000e-001, -1.40479e-002}
@COORDINATES { 8.70600e-001, -5.00000e-001, -1.55810e-002}
@COORDINATES { 9.62300e-001, -5.00000e-001, -1.71540e-002}
@COORDINATES { 1.05440e+000, -5.00000e-001, -1.87280e-002}
@COORDINATES {-5.46999e-002, -5.00000e-001, -1.13139e-002}
@COORDINATES { 3.38001e-002, -5.00000e-001, -1.25662e-002}
@COORDINATES { 1.22200e-001, -5.00000e-001, -1.38507e-002}
@COORDINATES { 2.10800e-001, -5.00000e-001, -1.51387e-002}
@COORDINATES { 3.03400e-001, -5.00000e-001, -1.65116e-002}
@COORDINATES { 3.99300e-001, -5.00000e-001, -1.78316e-002}
@COORDINATES { 4.93900e-001, -5.00000e-001, -1.88946e-002}
@COORDINATES { 5.88300e-001, -5.00000e-001, -2.02673e-002}
@COORDINATES { 6.83100e-001, -5.00000e-001, -2.21640e-002}
@COORDINATES { 7.78800e-001, -5.00000e-001, -2.38959e-002}
@COORDINATES { 8.73500e-001, -5.00000e-001, -2.54363e-002}
@COORDINATES { 9.66500e-001, -5.00000e-001, -2.70275e-002}
@COORDINATES { 1.05980e+000, -5.00000e-001, -2.86771e-002}
@COORDINATES {-5.84002e-002, -5.00000e-001, -2.10100e-002}
@COORDINATES { 3.06997e-002, -5.00000e-001, -2.22410e-002}
@COORDINATES { 1.19900e-001, -5.00000e-001, -2.35032e-002}
@COORDINATES { 2.10400e-001, -5.00000e-001, -2.47920e-002}
@COORDINATES { 3.04100e-001, -5.00000e-001, -2.61257e-002}
@COORDINATES { 4.00100e-001, -5.00000e-001, -2.73453e-002}
@COORDINATES { 4.94400e-001, -5.00000e-001, -2.80820e-002}
@COORDINATES { 5.88800e-001, -5.00000e-001, -2.92760e-002}
@COORDINATES { 6.83400e-001, -5.00000e-001, -3.15023e-002}
@COORDINATES { 7.79000e-001, -5.00000e-001, -3.35553e-002}
@COORDINATES { 8.74400e-001, -5.00000e-001, -3.51408e-002}
@COORDINATES { 9.68700e-001, -5.00000e-001, -3.67805e-002}
@COORDINATES { 1.06310e+000, -5.00000e-001, -3.84962e-002}
@COORDINATES {-6.14996e-002, -5.00000e-001, -3.06351e-002}
@COORDINATES { 2.81000e-002, -5.00000e-001, -3.18515e-002}
@COORDINATES { 1.18200e-001, -5.00000e-001, -3.30803e-002}
@COORDINATES { 2.09900e-001, -5.00000e-001, -3.43059e-002}
@COORDINATES { 3.04000e-001, -5.00000e-001, -3.54684e-002}
@COORDINATES { 3.99300e-001, -5.00000e-001, -3.64211e-002}
@COORDINATES { 4.94200e-001, -5.00000e-001, -3.71876e-002}
@COORDINATES { 5.89100e-001, -5.00000e-001, -3.83998e-002}
@COORDINATES { 6.84200e-001, -5.00000e-001, -4.04461e-002}
@COORDINATES { 7.79600e-001, -5.00000e-001, -4.26845e-002}
@COORDINATES { 8.75100e-001, -5.00000e-001, -4.45900e-002}
@COORDINATES { 9.70100e-001, -5.00000e-001, -4.63504e-002}
@COORDINATES { 1.06490e+000, -5.00000e-001, -4.81319e-002}
@COORDINATES {-6.43997e-002, -5.00000e-001, -4.02327e-002}
@COORDINATES { 2.54002e-002, -5.00000e-001, -4.14353e-002}
@COORDINATES { 1.15900e-001, -5.00000e-001, -4.26324e-002}
@COORDINATES { 2.08100e-001, -5.00000e-001, -4.37783e-002}
@COORDINATES { 3.02700e-001, -5.00000e-001, -4.47670e-002}
@COORDINATES { 3.98300e-001, -5.00000e-001, -4.54886e-002}
@COORDINATES { 4.94000e-001, -5.00000e-001, -4.62122e-002}
@COORDINATES { 5.89300e-001, -5.00000e-001, -4.74790e-002}
@COORDINATES { 6.84700e-001, -5.00000e-001, -4.94743e-002}
@COORDINATES { 7.80400e-001, -5.00000e-001, -5.17988e-002}
@COORDINATES { 8.76100e-001, -5.00000e-001, -5.39467e-002}
```

```
    @COORDINATES { 9.71300e-001, -5.00000e-001, -5.58447e-002}
    @COORDINATES { 1.06630e+000, -5.00000e-001, -5.76901e-002}
  }
  @KNOT_SEQUENCE_DEFINITION {
    @KNOT_SEQUENCE {
     0.00000e+000,  0.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e-001,
      2.00000e-001,
     3.00000e-001,  4.00000e-001,  5.00000e-001,  6.00000e-001,  7.00000e-001,
      8.00000e-001,
     9.00000e-001,  1.00000e+000,  1.00000e+000,  1.00000e+000,  1.00000e+000}
    @KNOT_SEQUENCE {
     0.00000e+000,  0.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e-001,
      2.00000e-001,
     3.00000e-001,  4.00000e-001,  5.00000e-001,  6.00000e-001,  7.00000e-001,
      8.00000e-001,
     9.00000e-001,  1.00000e+000,  1.00000e+000,  1.00000e+000,  1.00000e+000}
  }
}
@SURFACE_NAME {surface_blade_tip} {
  @IS_DEFINED_IN_FRAME {INERTIAL}
  @POINT_DEFINITION {
    @NUMBER_OF_CONTROL_POINTS {     4,      5}
    @DEGREE_OF_CURVE {     3,      3}
    @RATIONAL_CURVE_FLAG {NO}
    @COORDINATES { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    @COORDINATES { 3.33004e-002, -1.66589e-001, -9.24754e-018}
    @COORDINATES { 6.67000e-002, -3.33283e-001, -1.85009e-017}
    @COORDINATES { 1.00000e-001, -5.00000e-001, -2.77556e-017}
    @COORDINATES { 9.00000e-001,  1.89950e-016,  1.00000e-001}
    @COORDINATES { 8.33900e-001, -1.65629e-001,  8.89569e-002}
    @COORDINATES { 7.65400e-001, -3.35377e-001,  7.76415e-002}
    @COORDINATES { 7.00000e-001, -5.00000e-001,  6.66667e-002}
    @COORDINATES { 1.50000e+000, -2.05189e-015,  0.00000e+000}
    @COORDINATES { 1.36770e+000, -1.64990e-001, -1.86285e-019}
    @COORDINATES { 1.23130e+000, -3.36773e-001, -3.83195e-017}
    @COORDINATES { 1.10000e+000, -5.00000e-001, -2.77556e-017}
    @COORDINATES { 9.00000e-001,  1.89950e-016, -1.00000e-001}
    @COORDINATES { 8.33900e-001, -1.65629e-001, -8.89569e-002}
    @COORDINATES { 7.65400e-001, -3.35377e-001, -7.76415e-002}
    @COORDINATES { 7.00000e-001, -5.00000e-001, -6.66667e-002}
    @COORDINATES { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    @COORDINATES { 3.33004e-002, -1.66589e-001, -9.24754e-018}
    @COORDINATES { 6.67000e-002, -3.33283e-001, -1.85009e-017}
    @COORDINATES { 1.00000e-001, -5.00000e-001, -2.77556e-017}
  }
  @KNOT_SEQUENCE_DEFINITION {
    @KNOT_SEQUENCE {
     0.00000e+000,  0.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000,
     1.00000e+000,
     1.00000e+000,  1.00000e+000}
    @KNOT_SEQUENCE {
     0.00000e+000,  0.00000e+000,  0.00000e+000,  0.00000e+000,  2.69346e-001,
      1.00000e+000,
     1.00000e+000,  1.00000e+000,  1.00000e+000}
  }
}
@SURFACE_NAME {surface_swashplate} {
  @IS_DEFINED_IN_FRAME {INERTIAL}
  @POINT_DEFINITION {
    @NUMBER_OF_CONTROL_POINTS {     9,      5}
    @DEGREE_OF_CURVE {     2,      2}
    @RATIONAL_CURVE_FLAG {YES}
    @COORDINATES {-1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-1.00005e+000,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES {-1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-1.00005e+000,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES {-1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-1.00005e+000,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES {-1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-1.00005e+000,  0.00000e+000,  0.00000e+000,  7.07107e-001}
```

```
    @COORDINATES {-1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-1.00005e+000,  0.00000e+000,  1.82159e-001,  7.07107e-001}
    @COORDINATES {-1.00000e+000, -1.00000e+000,  1.82159e-001,  5.00000e-001}
    @COORDINATES {-1.00005e+000, -1.00000e+000,  5.05591e-017,  7.07107e-001}
    @COORDINATES {-1.00000e+000, -1.00000e+000, -1.82159e-001,  5.00000e-001}
    @COORDINATES {-1.00005e+000, -1.66533e-016, -1.82159e-001,  7.07107e-001}
    @COORDINATES {-1.00000e+000,  1.00000e+000, -1.82159e-001,  5.00000e-001}
    @COORDINATES {-1.00005e+000,  1.00000e+000,  2.02237e-017,  7.07107e-001}
    @COORDINATES {-1.00000e+000,  1.00000e+000,  1.82159e-001,  5.00000e-001}
    @COORDINATES {-1.00005e+000,  0.00000e+000,  1.82159e-001,  7.07107e-001}
    @COORDINATES { 0.00000e+000,  0.00000e+000,  1.82159e-001,  1.00000e+000}
    @COORDINATES {-5.63082e-005, -1.00000e+000,  1.82159e-001,  7.07107e-001}
    @COORDINATES { 0.00000e+000, -1.00000e+000,  5.05592e-017,  1.00000e+000}
    @COORDINATES {-5.63082e-005, -1.00000e+000, -1.82159e-001,  7.07107e-001}
    @COORDINATES { 0.00000e+000, -1.66533e-016, -1.82159e-001,  1.00000e+000}
    @COORDINATES {-5.63082e-005,  1.00000e+000, -1.82159e-001,  7.07107e-001}
    @COORDINATES { 0.00000e+000,  1.00000e+000,  2.02237e-017,  1.00000e+000}
    @COORDINATES {-5.63082e-005,  1.00000e+000,  1.82159e-001,  7.07107e-001}
    @COORDINATES { 0.00000e+000,  0.00000e+000,  1.82159e-001,  1.00000e+000}
    @COORDINATES { 9.99934e-001,  0.00000e+000,  1.82159e-001,  7.07107e-001}
    @COORDINATES { 1.00000e+000, -1.00000e+000,  1.82159e-001,  5.00000e-001}
    @COORDINATES { 9.99934e-001, -1.00000e+000,  5.05591e-017,  7.07107e-001}
    @COORDINATES { 1.00000e+000, -1.00000e+000, -1.82159e-001,  5.00000e-001}
    @COORDINATES { 9.99934e-001, -1.66533e-016, -1.82159e-001,  7.07107e-001}
    @COORDINATES { 1.00000e+000,  1.00000e+000, -1.82159e-001,  5.00000e-001}
    @COORDINATES { 9.99934e-001,  1.00000e+000,  2.02237e-017,  7.07107e-001}
    @COORDINATES { 1.00000e+000,  1.00000e+000,  1.82159e-001,  5.00000e-001}
    @COORDINATES { 9.99934e-001,  0.00000e+000,  1.82159e-001,  7.07107e-001}
    @COORDINATES { 1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES { 9.99934e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES { 1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES { 9.99934e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES { 1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES { 9.99934e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES { 1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES { 9.99934e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES { 1.00000e+000,  0.00000e+000,  0.00000e+000,  1.00000e+000}
  }
  @KNOT_SEQUENCE_DEFINITION {
    @KNOT_SEQUENCE {
     0.00000e+000,  0.00000e+000,  0.00000e+000,  2.50000e-001,  2.50000e-001,
     5.00000e-001,
     5.00000e-001,  7.50000e-001,  7.50000e-001,  1.00000e+000,  1.00000e+000,
      1.00000e+000}
    @KNOT_SEQUENCE {
     0.00000e+000,  0.00000e+000,  0.00000e+000,  5.00000e-001,  5.00000e-001,
      1.00000e+000,
     1.00000e+000,  1.00000e+000}
  }
}
@SURFACE_NAME {surface_nacelle} {
  @IS_DEFINED_IN_FRAME {INERTIAL}
  @POINT_DEFINITION {
    @NUMBER_OF_CONTROL_POINTS {     9,      5}
    @DEGREE_OF_CURVE {     2,      2}
    @RATIONAL_CURVE_FLAG {YES}
    @COORDINATES {-5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-5.00009e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES {-5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-5.00009e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES {-5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-5.00009e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES {-5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-5.00009e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
    @COORDINATES {-5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    @COORDINATES {-5.00009e-001,  0.00000e+000,  1.00000e+000,  7.07107e-001}
    @COORDINATES {-5.00000e-001, -3.33334e-001,  1.00000e+000,  5.00000e-001}
    @COORDINATES {-5.00009e-001, -3.33333e-001,  2.77556e-016,  7.07107e-001}
    @COORDINATES {-5.00000e-001, -3.33334e-001, -1.00000e+000,  5.00000e-001}
    @COORDINATES {-5.00009e-001, -5.55111e-017, -1.00000e+000,  7.07107e-001}
```

```
      @COORDINATES {-5.00000e-001,  3.33334e-001, -1.00000e+000,  5.00000e-001}
      @COORDINATES {-5.00009e-001,  3.33333e-001,  1.11022e-016,  7.07107e-001}
      @COORDINATES {-5.00000e-001,  3.33334e-001,  1.00000e+000,  5.00000e-001}
      @COORDINATES {-5.00009e-001,  0.00000e+000,  1.00000e+000,  7.07107e-001}
      @COORDINATES { 0.00000e+000,  0.00000e+000,  1.00000e+000,  1.00000e+000}
      @COORDINATES { 1.68587e-007, -3.33333e-001,  1.00000e+000,  7.07107e-001}
      @COORDINATES { 0.00000e+000, -3.33333e-001,  2.77556e-016,  1.00000e+000}
      @COORDINATES { 1.68587e-007, -3.33333e-001, -1.00000e+000,  7.07107e-001}
      @COORDINATES { 0.00000e+000, -5.55112e-017, -1.00000e+000,  1.00000e+000}
      @COORDINATES { 1.68587e-007,  3.33333e-001, -1.00000e+000,  7.07107e-001}
      @COORDINATES { 0.00000e+000,  3.33333e-001,  1.11022e-016,  1.00000e+000}
      @COORDINATES { 1.68587e-007,  3.33333e-001,  1.00000e+000,  7.07107e-001}
      @COORDINATES { 0.00000e+000,  0.00000e+000,  1.00000e+000,  1.00000e+000}
      @COORDINATES { 4.99995e-001,  0.00000e+000,  1.00000e+000,  7.07107e-001}
      @COORDINATES { 5.00000e-001, -3.33334e-001,  1.00000e+000,  5.00000e-001}
      @COORDINATES { 4.99995e-001, -3.33333e-001,  2.77556e-016,  7.07107e-001}
      @COORDINATES { 5.00000e-001, -3.33334e-001, -1.00000e+000,  5.00000e-001}
      @COORDINATES { 4.99995e-001, -5.55111e-017, -1.00000e+000,  7.07107e-001}
      @COORDINATES { 5.00000e-001,  3.33334e-001, -1.00000e+000,  5.00000e-001}
      @COORDINATES { 4.99995e-001,  3.33333e-001,  1.11022e-016,  7.07107e-001}
      @COORDINATES { 5.00000e-001,  3.33334e-001,  1.00000e+000,  5.00000e-001}
      @COORDINATES { 4.99995e-001,  0.00000e+000,  1.00000e+000,  7.07107e-001}
      @COORDINATES { 5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
      @COORDINATES { 4.99995e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
      @COORDINATES { 5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
      @COORDINATES { 4.99995e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
      @COORDINATES { 5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
      @COORDINATES { 4.99995e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
      @COORDINATES { 5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
      @COORDINATES { 4.99995e-001,  0.00000e+000,  0.00000e+000,  7.07107e-001}
      @COORDINATES { 5.00000e-001,  0.00000e+000,  0.00000e+000,  1.00000e+000}
    }
    @KNOT_SEQUENCE_DEFINITION {
      @KNOT_SEQUENCE {
       0.00000e+000,  0.00000e+000,  0.00000e+000,  2.50000e-001,  2.50000e-001,
        5.00000e-001,
       5.00000e-001,  7.50000e-001,  7.50000e-001,  1.00000e+000,  1.00000e+000,
        1.00000e+000}
      @KNOT_SEQUENCE {
       0.00000e+000,  0.00000e+000,  0.00000e+000,  5.00000e-001,  5.00000e-001,
        1.00000e+000,
       1.00000e+000,  1.00000e+000}
    }
  }
}
```

E.4.11 grfparameters.dat

```
@GRAPHICAL_PARAMETERS_DEFINITION {
  @GRAPHICAL_PARAMETERS_NAME {GrfParamBlade} {
    @REPRESENTATION_TYPE {MESH}
    @COLOR_FOR_CONFIGURATION {   255,      0,      0}
  }
  @GRAPHICAL_PARAMETERS_NAME {GrfParamFlexBeam} {
    @REPRESENTATION_TYPE {MESH}
    @COLOR_FOR_CONFIGURATION {     0,    255,      0}
  }
  @GRAPHICAL_PARAMETERS_NAME {GrfParamTube} {
    @REPRESENTATION_TYPE {SURFACE}
    @COLOR_FOR_CONFIGURATION {     0,      0,    255}
  }
  @GRAPHICAL_PARAMETERS_NAME {GrfParamHub} {
    @REPRESENTATION_TYPE {SURFACE}
    @COLOR_FOR_CONFIGURATION {   255,      0,    255}
  }
  @GRAPHICAL_PARAMETERS_NAME {GrfParamShaft} {
    @REPRESENTATION_TYPE {SURFACE}
    @COLOR_FOR_CONFIGURATION {     0,      0,    128}
  }
```

```
  @GRAPHICAL_PARAMETERS_NAME {GrfParamPitchlink} {
    @REPRESENTATION_TYPE {SURFACE}
    @COLOR_FOR_CONFIGURATION {   155,     48,    255}
  }
  @GRAPHICAL_PARAMETERS_NAME {GrfParamArbShaft2Pitchlink} {
    @REPRESENTATION_TYPE {SURFACE}
    @COLOR_FOR_CONFIGURATION {     0,    100,      0}
  }
  @GRAPHICAL_PARAMETERS_NAME {GrfParamArbHub2Tube} {
    @REPRESENTATION_TYPE {SURFACE}
    @COLOR_FOR_CONFIGURATION {   255,    255,      0}
  }
  @GRAPHICAL_PARAMETERS_NAME {GrfParamArbHub2Flexbeam} {
    @REPRESENTATION_TYPE {MESH}
    @COLOR_FOR_CONFIGURATION {     0,    255,    255}
  }
  @GRAPHICAL_PARAMETERS_NAME {GrfParamArbTube2Blade} {
    @REPRESENTATION_TYPE {TRIAD}
    @COLOR_FOR_CONFIGURATION {   255,    215,      0}
  }
  @GRAPHICAL_PARAMETERS_NAME {GrfParamArbTube2Pitchlink} {
    @REPRESENTATION_TYPE {SURFACE}
    @COLOR_FOR_CONFIGURATION {   255,    165,      0}
  }
}
```

E.5 DYMORE Input Files for Dynamic Analysis

E.5.1 Main File

```
@PROCESS_CONTROL_DEFINITION {
  @UNIT_SYSTEM {US}
  @DEBUG_PRINT_FLAG {     3}
  @FINITE_ELEMENT_ANALYSIS {YES}
  @POST_PROCESSING_ANALYSIS {YES}
  @SIGNAL_PROCESSING_ANALYSIS {YES}
  @MANUAL_PATH {C:\dymore2_0\manual\}
  @FIGURES_PATH {FIGURES\}
}
@MODEL_DEFINITION {
@FIXED_FRAME_DEFINITION {
  @FIXED_FRAME_NAME {frame_blade1} {
    @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  1.00000e+000}
  }
}
@MOVING_FRAME_DEFINITION {
  @MOVING_FRAME_NAME {BladeAttachedFrameQ1Q} {
    @CONNECTED_TO_BODY { flexbeamQ1Q }
    @AT_POINT          { point_flex_rootQ1Q }
    @TRIAD_NAME        { TriadBladeQ1Q }
  }
}
@TRIAD_DEFINITION {
  @TRIAD_NAME {TriadBladeQ1Q} {
    @ORIENTATION_E2      {0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3      {0.00000e+000,  0.00000e+000,  1.00000e+000}
    @IS_DEFINED_IN_FRAME {frame_blade1}
  }
}
@GRAVITY_DEFINITION {
  @GRAVITY_NAME {GravityLoading} {
    @GRAVITY_VECTOR { 0.00000e+000,  0.00000e+000, 0.0}
    !-3.22000e+001}
    @TIME_FUNCTION_NAME {NULL}
  }
}
@TIME_FUNCTION_DEFINITION {
```

```
  @TIME_FUNCTION_NAME {control_rotation_shaft} {
    @TIME_FUNCTION_TYPE {USER_DEFINED}
    @TABLE_ENTRIES {
      @TIME { 0.00000e+000}  @FUNCTION_VALUE { 0.00000e+000}
      @TIME { 1.00000e+002}  @FUNCTION_VALUE { 3281.34}
    }
  }
  @TIME_FUNCTION_NAME {prj_control_input} {
    @TIME_FUNCTION_TYPE {HARMONIC}
    @TABLE_ENTRIES {
      @AMPLITUDE { 0.00000e+000 } @PERIOD { 0.00000e+000} @PHASE { 0.00000e+000}
      @AMPLITUDE { 0.00000e+000 } @PERIOD { 0} @PHASE { 0}
      @AMPLITUDE { 0.00000e+000 } @PERIOD { 0} @PHASE { 0.00000e+000}
!      @AMPLITUDE { 0.00000e+000 } @PERIOD { 0.00000e+000} @PHASE { 0.00000e+000}
!      @AMPLITUDE { 0.00000e+000 } @PERIOD { 1.88513e-001} @PHASE { 2.50000e-001}
!      @AMPLITUDE { 0.00000e+000 } @PERIOD { 1.88513e-001} @PHASE { 0.00000e+000}
    }
  }
}
@SENSOR_DEFINITION {
  @SENSOR_NAME {SensorEnergy} {
    @OBJECT_NAME {ENERGY}
  }
}
@INCLUDE_COMMAND {
  @INCLUDE_COMMAND_NAME {IncludeRotorDimensionPropertyData} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/dimension_prop.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeBladeProperties} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/properties.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeAerodynamicEffect} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/aerodynamic.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeSensorsDynamic} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/sensors_dynamic.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludePoint} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/points.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeHub} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/hub.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeFlexBlade} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/flexblade.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeTorqueTube} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/torquetube.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludePitchLink} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/pitchlink.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeShape} {
```

```
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/shapes.dat}
  }
  @INCLUDE_COMMAND_NAME {IncludeGrfParameters} {
    @ACTIVE_COMMAND {YES}
    @LIST_OF_FILE_NAMES {
    model/grfparameters.dat}
  }
}
}
@CREATE_FINITE_ELEMENT_MODEL {
@CFM_CONTROL_PARAMETERS {
  @ANALYSIS_TYPE {DYNAMIC}
}
}
@FINITE_ELEMENT_ANALYSIS {
@FEM_CONTROL_PARAMETERS {
  @FEM_CONTROL_PARAMETERS_NAME {AnalysisControlParameters} {
    @MAXIMUM_NUMBER_OF_TIME_STEPS {    180000}
    @SIMULATION_TIME_RANGE { 0.00000e+000,  15.00000e+0}
    @TIME_STEP_SIZE_RANGE { 1.00000e-004,  1.00000e-001}
    @REFERENCE_ENERGY_LEVEL { 3.00000e+007}
  }
}
@INITIAL_CONDITION_DEFINITION {
  @INITIAL_CONDITION_NAME {InitialConditionForOneBladeDynamicAnalysis} {
    @FILE_NAME {LHY_static.rcv}
    @FILE_TIME_STEP_NUMBER {    61}
    @INITIAL_TIME_STEP_NUMBER {     1}
    @INITIAL_TIME { 0.00000e+000}
  }
}
@STEP_CONTROL_PARAMETERS {
  @STEP_CONTROL_PARAMETER_NAME {Step1} {
    @ARCHIVAL_FREQUENCY {     10}
    @REUSE_NUMBER {360000}
    @SPECTRAL_RADIUS_AT_INFINITY { 0.00000e+000}
    @TIME_STEP_SIZE { 1.0e-001}
    @MAXIMUM_NUMBER_OF_ITERATIONS {    20}
    @FACTORIZATION_STRATEGY {     1}
    @CONVERGENCE_TOLERANCE { 1.00000e-06}
    @MAXIMUM_NUMBER_OF_REJECT {     5}
    @AVERAGE_STIFFNESS_TERM { 1.00000e+006}
    @AVERAGE_MASS_TERM { 0.00000e+000}
  }
}
}
@POST_PROCESSING_ANALYSIS {
@GRAPHICS_CONTROL_PARAMETERS {
  @GRAPHICS_CONTROL_PARAMETERS_NAME {GraphicsControlParameters} {
    @TIME_STEP_SIZE { 0.00000e+000}
    @EIGENVECTORS_SCALING_FACTOR { 2.00000e-001}
    @MODAL_ANIMATION_CYCLES {     5}
    @MODAL_ANIMATION_FRAMES_PER_CYCLE {    50}
    @VECTOR_FIELD_TYPE {FORCES}
    @VECTOR_FIELD_SCALING_FACTOR { 2.00000e-001}
    @VECTOR_FIELD_TYPE {VELOCITIES}
    @VECTOR_FIELD_SCALING_FACTOR { 2.00000e-001}
  }
}
@VIEW_PARAMETERS_DEFINITION {
  @VIEW_PARAMETERS_NAME {ViewParameters} {
    @VIEW_REFERENCE_POINT { 5.00000e-001,  5.00000e-001,  5.00000e-001}
    @VIEW_SIZE { 1.20000e+000,  1.20000e+000,  1.20000e+000}
    @PROJECTION_REFERENCE_POINT {-8.15283e-001, -8.24033e-001, -3.88000e-001}
    @PROJECTION_EYE_VECTOR { 0.00000e+000,  0.00000e+000,  1.00000e+000}
    @PROJECTION_UP_VECTOR { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @PROJECTION_VIEWPORT {-3.82063e-001,  3.82063e-001, -2.62535e-001,  2.62535e-001}
  }
}
```

```
}
```

E.5.2 aerodynamic.dat

```
@AERODYNAMIC_INTERFACE_DEFINITION {
  @AERODYNAMIC_INTERFACE_NAME {AeroIntName} {
    @AIRLOADS_SCHEME {2D_AIRFOIL}
    @AIR_PROPERTY_NAME {PropertyAir}
    @ROTOR_LIST {Rotor}
    @FRAME_DEFINITION {
       @FUSELAGE_FRAME_NAME { FrameFuselage }
    }
  }
}
@AIR_PROPERTY_DEFINITION {
@AIR_PROPERTY_NAME{PropertyAir}{
  @AIR_DENSITY {0.0023769}
  @SPEED_OF_SOUND          {1115.485564304461942257217847769}
  @FAR_FIELD_FLOW_VELOCITY {0.0, 0.0, 0.0}
  @SLOPE_OF_LIFT_CURVE     {6.5131}
  @DRAG_COEFFICIENTS        {0.013}
  @MOMENT_COEFFICIENTS     {0.0, 0.0}
  @GUST_VELOCITY           {100.0,0.0,0.0}
  @TIME_FUNCTION_NAME{gust_time}
  @PERIOD {3}  }}
  @TIME_FUNCTION_DEFINITION {
  @TIME_FUNCTION_NAME {gust_time} {
    @TIME_FUNCTION_TYPE {USER_DEFINED}
    @TABLE_ENTRIES {
      @TIME { 0.00000e+000}  @FUNCTION_VALUE { 0.00000e+000}
      @TIME { 5.00000}  @FUNCTION_VALUE { 0.00000e+000}
      @TIME { 5.0100000}  @FUNCTION_VALUE { 1.00000e+000}
      @TIME { 6.0100000}  @FUNCTION_VALUE { 1.00000e+000}
      @TIME { 6.020000}  @FUNCTION_VALUE { 0.00000e+000}
      @TIME { 1.00000e+002}  @FUNCTION_VALUE { 0.00000e+000}
    }
  }
}
@ROTOR_DEFINITION {
    @ROTOR_NAME { Rotor }{
      @NUMBER_OF_BLADES { 4 }
      @ROTOR_SPEED { 32.8134 }
      @ROTOR_RADIUS { 14.1673 }
      @SHAFT_FRAME_NAME { FrameShaft }
      @LIFTING_LINE_LIST { LfnLineQ1Q }
      @INFLOW_NAME { InflowDynamic }
      @REFERENCE_LENGTH { 14.1673 }
      @REFERENCE_CHORD_LENGTH { 0.999158333333333 }
   }
}
@INFLOW_DEFINITION {
  @INFLOW_NAME {InflowDynamic} {
    @INFLOW_TYPE{DYNAMIC_INFLOW}
    @NUMBER_OF_MODES {7}
  }
}
@FIXED_FRAME_DEFINITION {
  @FIXED_FRAME_NAME { FrameFuselage} {
    @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  -1.00000e+000}
  }
}
@FIXED_FRAME_DEFINITION {
  @FIXED_FRAME_NAME {FrameShaft} {
    @ORIGIN { 0.00000e+000,  0.00000e+000,  0.00000e+000}
    @ORIENTATION_E2 { 0.00000e+000,  1.00000e+000,  0.00000e+000}
    @ORIENTATION_E3 { 0.00000e+000,  0.00000e+000,  -1.00000e+000}
```

```
  }
}
@LIFTING_LINE_DEFINITION {
  @LIFTING_LINE_NAME {LfnLineQ1Q} {
    @IS_DEFINED_IN_FRAME {FrameBladeSweepQ1Q}
    @AIRSTATION_LOCATION_SCHEME {EQUALLY_SPACED}
    @LIFTING_LINE_PROPERTY_NAME {PropertyLfnLine}
    @ORIENTATION_DISTRIBUTION_NAME {OriDistLfnLine}
    @NUMBER_OF_AIRSTATIONS {   20  }
    @BODY_LIST {bladeQ1Q}
    @INITIAL_POINT {0.00e+000, 0.0, 0.0}
    @CURVILINEAR_COORDINATE { 0.0, 9.76 }
    @TIP_LOSS_FACTOR{0.97}
    @GRAPHICAL_PARAMETERS_NAME {GrfParamLfnline}
  }
}
@GRAPHICAL_PARAMETERS_DEFINITION {
  @GRAPHICAL_PARAMETERS_NAME {GrfParamLfnline} {
    @REPRESENTATION_TYPE {LINE}
    @COLOR_FOR_CONFIGURATION {     0,    255,    255}
    @VECTOR_FIELD_TYPE {AERODYNAMIC_FORCES}
    @COLOR_FOR_VECTOR_FIELD {     0,      0,    255}
  }
}
@ORIENTATION_DISTRIBUTION_DEFINITION{
  @ORIENTATION_DISTRIBUTION_NAME{OriDistLfnLine}{
    @CURVILINEAR_COORDINATE {0.00000e+000}
    @EULER_ANGLES_313 {0.00000e+000,1.20000e+001,0.00000e+000}
    @CURVILINEAR_COORDINATE { 9.76 }
    @EULER_ANGLES_313 {0.00000e+000,5.00000e+000,0.00000e+000}
  }
}

@LIFTING_LINE_PROPERTY_DEFINITION {
  @LIFTING_LINE_PROPERTY_NAME {PropertyLfnLine} {
    @CHORD_LENGTH_DEFINITION {
     @CURVILINEAR_COORDINATE{0.00}
      @CHORD_LENGTH {0.999158333333333}
      @QUARTER_CHORD_OFFSET{0.00}
    }
 }
}
```

E.5.3 sensors_dynamic.dat

```
@SENSOR_DEFINITION {
  @SENSOR_NAME {SensLfnLineFlowr} {
    @OBJECT_NAME {LfnLineQ1Q}
    @SENSOR_TYPE {FLOW_PARAMETERS}
    @ETA_VALUE { 1.00000e+000}
  }
  @SENSOR_NAME {SensLfnLineFlowt} {
    @OBJECT_NAME {LfnLineQ1Q}
    @SENSOR_TYPE {FLOW_PARAMETERS}
    @ETA_VALUE { 2.00000e+001}
  }
  @SENSOR_NAME {SensLfnLineForce20} {
    @OBJECT_NAME {LfnLineQ1Q}
    @SENSOR_TYPE {AIRSTATION_LOADS}
    @ETA_VALUE { 2.00000e+001}
  }
  @SENSOR_NAME {SensorRotorLoads} {
    @OBJECT_NAME {Rotor}
    @SENSOR_TYPE {TOTAL_AIRLOADS}
  }
  @SENSOR_NAME {SensFlexBeam1For00} {
    @OBJECT_NAME {flexbeamQ1Q}
    @SENSOR_TYPE {FORCES}
```

```
    @ETA_VALUE { 0.00000e+000}
  }
  @SENSOR_NAME {SensFlexBeam1For10} {
    @OBJECT_NAME {flexbeamQ1Q}
    @SENSOR_TYPE {FORCES}
    @ETA_VALUE { 1.00000e+000}
  }
  @SENSOR_NAME {SensBeam1For00} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {FORCES}
    @ETA_VALUE { 0.00000e-000}
  }
  @SENSOR_NAME {SensBeam1For050} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {FORCES}
    @ETA_VALUE { 0.50000e+000}
  }
  @SENSOR_NAME {SensBeam1For100} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {FORCES}
    @ETA_VALUE { 1.00000e+000}
  }
  @SENSOR_NAME {SensBeam1Dis06} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {DISPLACEMENTS}
    @ETA_VALUE { 0.60000e+000}
  }
  @SENSOR_NAME {SensBeam1Posi060} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {POSITIONS}
    @ETA_VALUE { 0.60000e+000}
  }
  @SENSOR_NAME {SensorBladeDisp050} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {DISPLACEMENTS}
    @ETA_VALUE { 5.00000e-001}
  }
  @SENSOR_NAME {SensorBladeDisp075} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {DISPLACEMENTS}
    @ETA_VALUE { 7.50000e-001}
  }
  @SENSOR_NAME {SensorBladeDisp100} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {DISPLACEMENTS}
    @ETA_VALUE { 1.00000e+000}
  }
  @SENSOR_NAME {SensBeam1Dis000} {
    @OBJECT_NAME {bladeQ1Q}
    @SENSOR_TYPE {DISPLACEMENTS}
    @FRAME_NAME {BladeAttachedFrameQ1Q}
    @ETA_VALUE { 1.0}
  }
}
```

REFERENCES

[1] Technology for Rotorcraft Affordability through Integrated Product/Process Development (IPPD), Schrage, D.P., American Helicopter Society 55th Annual Forum, Montreal, Canada, May 25-29, 1999

[2] Framework for Multidisciplinary Analysis, Design, and Optimization with High-Fidelity Analysis Tools, Orr, S.A., Narducci, R.P., NASA technical report no. NASA/CR-2009-215563, February 2009

[3] Development of a Digital Framework for Integrated Product/Process Development (IPPD), Chae, H.G., Gunduz, M.E., Sirirojvisuth A., Liu H.Y., Schrage, D.P., AHS 2nd International Forum on Rotorcraft Multidisciplinary Technology, Seoul, Korea, October 2009

[4] Engineering Analysis of the 1907 Cornu Helicopter, Leishmann, J.G., Johnson, B., 64th Annual Forum of the American Helicopter Society, Montreal, Canada, http://helicopter-history.org/Cornu/Cornu_LJpaper.pdf, accessed on Mar 20th, 2010

[5] A Summary of Industry MDO Applications and Needs, Giesing, J.P., Barthelemy, J.F.M., AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 7th, St. Louis, MO, Sept. 2-4, 1998, AIAA-1998-4737

[6] IPPD Through Robust Design Simulation for an Affordable Short Haul Civil Tiltrotor, Mavris, D., Baker, A., Schrage, D., Proceedings of the American Helicopter Society 53rd Annual Forum, Virginia Beach, VA, April 29-May 1 1997

[7] The Configuration and Conceptual Design for Rotary Wing Aircraft, Bates, P.R., Schrage, D.P., Proceedings of the AIAA/AHS/ASEE Aircraft Design Systems and Operations Meeting, St. Louis, Missouri, USA, September 14-16, 1987, pp. 10

[8] Engineering Design Handbook, Helicopter Engineering, Part One: Preliminary Design, Headquarters U.S. Army Materiel Command, Alexandria, VA. 22333, AMC 706-201, pp. 3-90,91 August 1974

[9] Integrating Design and Manufacturing for the High Speed Civil Transport, Marx, W.J., Mavris, D.N., Schrage, D.P., American Institute of Aeronautics and Astronautics, Inc., September 1994

[10] Methodology for Examining The Simultaneous Impact of Requirements, Vehicle Characteristics, and Technologies on Military Aircraft Design, Mavris, D.N., DeLaurentis, D., ICAS 145.1

[11] Advanced Design Methods I course slides, Mavris, D.N., Georgia Institute of Technology, School of Aerospace Engineering, 2004

[12] Current state of the Art on Multidisciplinary Design Optimization (MDO), White Paper, AIAA Technical Committee, January 15th 1991, ISBN 1563470217

[13] A Methodology for Technology Identification, Evaluation, and Selection in conceptual and preliminary aircraft design, Michelle R. Kirby, Ph.D. Dissertation, Georgia Institute of Technology, 2001

[14] Recomposition: The Other Half of the MDO Equation, Schrage, D.P., Mavris, D.N., Proceedings of the ICASE/NASA Langley Workshop in Multi-disciplinary Design Optimization - State-of-the-Art, March 13-16, 1995

[15] IPPD Through Robust Design Simulation for an Affordable Short Haul Civil Tiltrotor, Mavris, D., Baker, A., Schrage, D., Proceedings of the American Helicopter Society 53rd Annual Forum, Virginia Beach, VA, April 29-May 1 1997

[16] A Generalized Graphical Method of Minimum Gross Weight Estimation, Simonds, R. M., Presented at the national conference of the society of aeronautical weight engineers, INC. San Diego, CA. May 1956

[17] CATIA V5 / ENOVIA LCA Relational Design Workshop Job Aid Booklet, BOEING Learning, Training and Development group, 2007

[18] Linear Flap-Lag Dynamics of Hingeless Helicopter Rotor Blades in Hover, Ormiston, R.A., Hodges, D.H., J. American Helicopter Soc., vol. 17, no. 2, Apr. 1972, pp. 2 - 14

[19] Recent Applications of Design Optimization to Rotorcraft - A Survey, Celi, R., Journal of Aircraft, vol.36, n.1, Jan-Feb, 1999, pp. 176-189

[20] Techniques for Stability Analysis on Design Optimization with Dynamic Constraints on Non Conservative Linear Systems, Bielawa, R.L., AIAA Paper 71-388, 1971

[21] Application of Optimization Methods to Rotor Design Problems, Bennet, R.L., Vertica, vol. 7, n. 3, 1983, pp. 201-208

[22] Optimum Design of Rotor Blades for Vibration Reduction in Forward Flight, Friedmann, P.P., Shantakumaran, P., Journal of the American Helicopter Society, vol. 29, n. 4, Oct. 1984, pp. 70-80

[23] Optimization of Helicopter Blade Design for Minimum Vibration, Davis, M.W., NASA CP-2327, Sep. 1984, pp. 609-625

[24] Design of Helicopter Rotor Blades for Desired Placement of Natural Frequencies, Peters, D.A., Ko, T., Korn, A., Rossow, M.P., Proceedings of the 39th Annual Forum of the American Helicopter Society, Anaheim, CA, May 1983, pp. 674-689

[25] Non-Linear Flap-Lag Dynamics of Hingeless Helicopter Blades in Hover and in Forward Flight, Friedmann, P., Tong, P., Journal of Sound and Vibration, Vol. 30, n. 1, pp. 9-31, 1973

[26] Aeroelastic Stability of Coupled Flap-Lag Motion of Hingeless Helicopter Blades at Arbitrary Advance Ratios, Friedmann, P., Silverthorn, L.J., Journal of Sound and Vibration, Vol. 39, n. 4, pp. 409-428, 1975

[27] Application of Modern Structural Optimization to Vibration Reduction in Rotorcraft, Friedmann, R., Vertica, Vol. 9, n. 4, pp. 363-376, 1985

[28] The Finite Element for Engineers, Huebner, K.H., Wiley-IEEE, 4th ed. , 2001

[29] Aeroelastic Optimization of a Helicopter Rotor, Lim, J.W., Chopra, I., AHS, Annual Forum, 44th, Washington, DC, Proceedings; 16-18 June 1988. pp. 545-558. 1988

[30] Coupled Flap-Lag Torsional Dynamics of Hingeless Rotor Blades in Forward Flight, Friedmann, P.P., Kotapalli, S.B.R., Journal of the American Helicopter Society, Vol. 27, pp. 28-36, October 1982

[31] Structural Optimization with Aeroelastic Constraints of Rotor Blades with Straight and Swept Tips, Celi, R., Friedmann, P.P., AIAA Journal, Vol. 28, n. 1, pp. 928-936, 1990

[32] Integrated Multidisciplinary Design Optimization of Rotorcraft, Adelman, H.M., Mantay, W.R., Journal of Aircraft, Vol. 28, n. 1, pp. 22-28, 1991

[33] Integrated Aerodynamic Load/Dynamic Optimization of Helicopter Rotor Blades, Chattopadhyay, A., Walsh, J.L., Riley, M.F., Journal of Aircraft, Vol. 28, n. 1, pp. 58-65, 1991

[34] An Enhanced Integrated Aerodynamic / Dynamic Approach to Optimum Rotor Blade Design, Chattopadhyay, A., Chiu, Y.D., Structural Optimization, Vol. 4, pp., 75-84, 1992

[35] Helicopter Rotor Dynamics Optimization with Experimental Verification, Davis, M.W., Weller, W.H., Journal of Aircraft, Vol. 28 n. 1, pp. 38-48, Jan 1991

[36] Optimum Aeroelastic Design of Helicopter Rotors for Longitudinal Handling Qualities Improvement, Celi, R., Journal of Aircraft, Vol. 28, n. 1, pp. 49-57. Jan. 1991

[37] Aeroelastic Optimization of a Helicopter Rotor Using an Efficient Sensitivity Analysis, Lim, J.W., Chopra, I., Journal of Aircraft, Vol. 28, pp. 29-37. Jan. 1991

[38] Fully Integrated Aerodynamic / Dynamic Optimization of Helicopter Rotor Blades, Walsh, J.L., LaMarsh, W.J., Adelman, H.M., AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 33rd, Dallas, TX; UNITED STATES; pp. 2640-2655, 13-15 Apr. 1992

[39] A New Sensitivity Analysis for Structural Optimization of Composite Rotor Blades, Vankatesan C., Friedmann, P.P., Yuan, K., AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 34th and AIAA/ASME Adaptive Structures Forum, La Jolla, CA; USA; pp. 2952-2973, 19-22 Apr. 1993

[40] Aeroelastic Stability of Composite Hingeless Rotor Blades in Hover - Part I: Theory, Fulton, M.V., Hodges, D.H., Mathematical Computer Modelling, Vol. 18, n. , pp. 1-17, 1993

[41] Aeroelastic Stability of Composite Hingeless Rotor Blades in Hover - Part II: Results, Fulton, M.V., Hodges, D.H., Mathematical Computer Modelling, Vol. 18, n. , pp. 19-35, 1993

[42] Optimizing Tuning Masses for Helicopter Rotor Blade Vibration Reduction and Comparison with Test Data, Pritchard, J.I., Adelman, H.M., Walsh, J.L., Wilbur, M.L., Journal of Aircraft, Vol. 30, n. 6, pp. 906-910, 1993

[43] Optimization of Composite Rotor Blades with Advanced Structural and Aerodynamic Modeling, Barwey, D., Peters, D.A., Mathematical and computer modeling, Rotorcraft Modelling II, Vol. 19, n. 3-4, pp.193-219, 1994

[44] A New Sensitivity Analysis for Structural Optimization of Composite Rotor Blades, Vankatesan, C., Friedmann, P.P., Yuan, K.-A., Mathematical Computer Modelling, Vol. 19, n. , pp. 1-25, 1994

[45] Efficient Sensitivity Analysis for Rotary-Wing Aeromechanical Problems, Spence, A.M., Celi, R., AIAA Journal, Vol. 32, n. 12, pp. 2337-2344, 1994

[46] Multilevel Decomposition Approach to Integrated Aerodynamic / Dynamic / Structural Optimization of Helicopter Rotor Blades, Walsh, J.L., Young, K.C., Pritchard, J.I., Adelman, H.M., Mantay, W.R., NASA TM-109084, May 1994

[47] Integrated Aerodynamic / Dynamic / Structural Optimization of Helicopter Rotor Blades Using Multilevel Decomposition, Walsh, J.L., Young, K.C., Pritchard, J.I., Adelman, H.M., Mantay, W.R., NASA TP 3465, Jan 1995

[48] Application of Response Surface Techniques to Helicopter Rotor Blade Optimization Procedure, Henderson, J.L., Walsh, J.L., Young, K.C., Proceedings of the AHS National Technical Specialist Meeting on Rotorcraft Structures: Design Challenges and Innovative Solutions, 1995

[49] Aeroelastic Optimization of a Helicopter Rotor with Composite Coupling, Ganguli, R., Chopra, I., Journal of Aircraft, Vol. 32, n. 6, pp.1326-1334, November-December 1995

[50] Semi-Analytical Sensitivity of Floquet Characteristic Exponents with Application to Rotary Wing Aeroelasticity, Shih, I.-C., Spence, A.M., Celi, R., Journal of Aircraft, Vol. 33, n. 2, pp. 322-330, 1996

[51] Aeroelastic Optimization of a Helicopter Rotor to Reduce Vibration and Dynamic Stresses, Ganguli, R., Chopra, I., Journal of Aircraft, Vol. 12, n. 4, pp. 808-815, July-August 1996

[52] Aeroelastic Optimization of a Helicopter Rotor with Two-Cell Composite Blades, Ganguli, R., Chopra, I., AIAA Journal, Vol. 34, n. 4, pp. 835-841, April 1996

[53] Dynamics Workshop on Rotor Vibratory Loads Prediction, Hansford, R.E., Vorwald, J., Journal of the American Helicopter Society, Vol. 43, n. 1, pp. 76-87, 1998

[54] Optimized Aeroelastic Couplings for Alleviation of Helicopter Ground Resonance, Gandhi, F., Hathaway, E., Journal of Aircraft, Vol. 35, n. 4, pp. 582-589, 1998

[55] Concurrently Optimized Aeroelastic Couplings and Rotor Stiffness for Alleviation of Helicopter Aeromechanical Instability, Hathaway, E., Gandhi, F., Journal of Aircraft, Vol. 38, n. 1, pp. 69-80, 2001

[56] Optimum Design of a Helicopter Rotor for low Vibration Using Aeroelastic Analysis and Response Surface Methods, Ganguli, R., Journal of Sound and Vibration, Vol. 258, n. 2, pp. 327-344, 2002

[57] Survey of Recent Developments in Rotorcraft Design Optimization, Ganguli, R., Journal of Aircraft, Vol. 41, n. 3, pp. 493-510, May-June 2004

[58] Helicopter Vibration Reduction Using Structural Optimization with Aeroelastic/Multidisciplinary Constraints - A Survey, Friedmann, P.P., Journal of Aircraft, Vol. 28, n. 1, 1991, pp. 8-21

[59] Aeroelastic Stability Enhancement and Vibration Suppression in a Composite Helicopter Rotor, Murugan, S., Ganguli, R., Journal of Aircraft, Vol. 42, n. 4, pp. 1013-1024, July-August 2005

[60] Surrogate Based Optimization of Helicopter Rotor Blades for Vibration Reduction in Forward Flight, Glaz, B., Friedmann, P.P., Liu, L., 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; Newport, RI, USA, pp. 1-21, 1-4 May 2006.

[61] Aeroelastic Optimization of a Helicopter Rotor Using Orthogonal Array- Based Metamodels, Bhadra, S., Ganguli, R., AIAA Journal, Vol. 44, n. 9, pp. 1941-1951, September 2006

[62] A Hybrid Optimization Scheme for Helicopters with Composite Rotor Blades, Ku, J., PhD thesis, Georgia Institute of Technology, August 2007

[63] A multi-fidelity framework for physics based rotor blade simulation and optimization, Collins, K.B., PhD thesis, Georgia Institute of Technology, December 2008

[64] Pareto Frontier Method for Multi-Disciplinary Optimization of Helicopter Rotors, Collins, K., Bain, J., Sankar, L., Egolf, T.A., Janakiram, R.D., Brentner, K., Lopes, L., AHS Specialist's Conference on Aeromechanics, San Francisco, CA, Jan. 23-25, 2008

[65] Structural Synthesis - Its Genesis and Development, Schmit, L.A., AIAA Journal, vol. 19, n. 10, Oct 1981, pp. 1249-1263

[66] The Autotrim Rotor Stability System, Hanson, T.F., AHS 53rd Annual Forum, Virginia Beach, VA, 1997, pp. 1503

[67] Elastic Articulation with Composite Flexures, Hanson, T.F., AHS 53rd Annual Forum, Virginia Beach, VA, 1997, pp. 141

[68] The Simplicity of Elastic Articulation, Hanson, T.F., AHS 53rd Annual Forum, Virginia Beach, VA, 1997, pp. 545

[69] A Designer Friendly Handbook of Helicopter Rotor Hubs, Hanson, T.F., self published, 1998

[70] Efficient and Robust Approaches for Rotorcraft Stability Analysis, Bauchau, O.A., Wang, J., AHS 63rd Annual Forum Proceedings, 2007

[71] On a Simplified Strain Energy Function for Geometrically Nonlinear Behaviour of Anisotropic Beams, Hodges, D.H., Atilgan, A.R., Cesnik, C.E.S., Fulton, M.V., Composites Engineering, vol. 2, n. 5 - 7, pp. 513 - 526, 1992

[72] Validation of the Variational Asymptotic Beam Sectional Analysis (VABS), Yu, W., Volovoi, V.V., Hodges, D.H., and Hong, X., AIAA Journal, vol. 40, n. 10, pp. 2105-2112, 2002

[73] Variational Asymptotic Modeling of Composite Dimensionally Reducible Structures, Yu, W., Ph.D. thesis, Georgia Institute of Technology, 2002.

[74] On Timoshenko-Like Modeling of Initially Curved and Twisted Composite Beams, Yu, W., Hodges, D.H., Volovoi, V.V., Cesnik, C.E.S., International Journal of Solids and Structures, vol. 39, n. 19, pp. 5101-5121, 2002

[75] Elasticity Solutions Versus Asymptotic Sectional Analysis of Homogeneous, Isotropic, Prismatic Beams, Yu, W., Hodges, D.H., Journal of Applied Mechanics, vol. 71, n. 1, 2004, pp. 15-23, 2004

[76] Generalized Timoshenko Theory of the Variational Asymptotic Beam Sectional Analysis, Yu, W. Hodges, D.H., Journal of the American Helicopter Society, vol. 50, n. 1, pp. 46-55, 2005

[77] A Tutorial of VABS, Yu, W., 2005.

[78] DYMORE Theory Manual, Bauchau, O.A., 2002, http://soliton.ae.gatech.edu/people/obauchau/, accessed on Feb 22nd, 2008

[79] Energy decaying scheme for non-linear beam models, Bauchau, O.A., Theron, N.J., Computer Methods in Applied Mechanics and Engineering, vol. 134, pp. 37-56, 1996

[80] Integrating finite rotations, Bottasso, C.L., Borri, M. Computer Methods in Applied Mechanics and Engineering, vol. 164, pp. 307-331, 1998.

[81] On the design of energy preserving and decaying schemes for flexible, nonlinear multi-body systems, Bauchau, O.A., Bottasso, C.L., Computer Methods in Applied Mechanics and Engineering, vol. 169, pp. 61-79, 1999.

[82] Analysis of nonlinear multi-body systems with elastic couplings, Bauchau, O.A., Hodges, D.H., Multibody System Dynamics, vol. 3, pp. 168-188, 1999.

[83] Robust integration schemes for flexible multibody systems, Bauchau, O.A., Bottasso, C.L., Trainelli, L., Computer Methods in Applied Mechanics and Engineering, 2000.

[84] On the modeling of prismatic joints in flexible multi-body systems, Bauchau, O.A., Computer Methods in Applied Mechanics and Engineering, vol. 181, pp. 87-105, 2000.

[85] Contact conditions for cylindrical, prismatic, and screw joints in flexible multibody systems, Bauchau, O.A., Bottasso, C.L., Multibody System Dynamics, vol. 5, pp. 251-278, 2001.

[86] Analysis of flexible multi-body systems with intermittent contacts, Bauchau, O.A., Multibody System Dynamics, vol. 4, pp. 23-54, 2000.

[87] On the modeling of friction and rolling in flexible multi-body systems, Bauchau, O.A., Multibody System Dynamics, vol. 3, pp. 209-239, 1999.

[88] DYMORE: A Finite Element Based Tool for the Analysis of Nonlinear Flexible Multibody Systems; Bauchau, O.A., http://www.ae.gatech.edu/people/obauchau/dymore.pdf, accessed on Feb 22nd, 2008

[89] Structural Design by Systematic Synthesis, Schmit, L. A., Proceedings of 2nd Conference on Electronic Computation, American Society of Civil Engineers, New York, 1960, pp. 105-122

[90] Multi-variable search and its applications to aircraft design optimization, Stepniewski, W.Z., C.F. Kalmbac, J., Aeronautical Journal, Royal Aeronautical Society, Vol. 74, 1970.

[91] Approach to Multidisciplinary Design Optimization, Simpson, T.W., Presentation for Mechanical & Nuclear Engineering and Industrial & Manufacturing Engineering, The Pennsylvania State University, University Park, PA. 16802

[92] Development and Implementation of Rotorcraft Preliminary Design Using Multidisciplinary Design Optimization, Khalid, A.S., Ph.D. Thesis, Georgia Institute of Technology, School of Aerospace Engineering, December 2006

[93] Bramwell's Helicopter Dynamics, Bramwell, A.R.S., Done, G., Balmford, D., 2nd ed., Butterworth-Heinemann, 2001

[94] Composite Airframe Structures: Practical Design Information and Data, Niu, M.C., 4th ed., Conmilit Press, May 2005

[95] Aerospace Structural Analysis, Bauchau, O.A., Craig, J.I., January 2008, http://www.ae.gatech.edu/people/obauchau/Dwnld/classNotes/tos.pdf, accessed on Feb 22nd, 2008